

Universidad Carlos III de Madrid
Escuela Politécnica Superior
Ingeniería en Informática



Proyecto Fin de Carrera

**Desarrollo de una aplicación de comunicación
Bluetooth entre una PDA y sensores.**

Autor: D. David Barbolla Asenjo

Tutor: Prof. D. Javier Ortiz Laguna

Fecha: Julio 2009

Título: Desarrollo de una aplicación de comunicación Bluetooth entre una PDA y sensores

Asunto: Memoria del Proyecto Fin de Carrera de Ingeniería en Informática.

Autor: D. David Barbolla Asenjo

Tutor: Prof. D. Javier Ortiz Laguna

Universidad Carlos III de Madrid

Campus de Leganés

PROYECTO FIN DE CARRERA

Departamento de Inteligencia Artificial
Universidad Carlos III de Madrid

Título: Desarrollo de una aplicación de comunicación
Bluetooth entre una PDA y sensores

Autor: D. David Barbolla Asenjo

Tutor: D. Javier Ortiz Laguna

La lectura y defensa del presente Proyecto Fin de Carrera se realizó el día 16 de Julio de 2009 bajo el tribunal:

- **Presidente:** Dr. Daniel Borrajo Millán
- **Secretario:** D. Daniel Pérez Pinillos
- **Vocal:** Dr. Rodolfo Cuerno Rejado

Habiendo obtenido la calificación de:

Presidente

Secretario

Vocal

A mis padres José Antonio y Fuencis,
a mis hermanas Patricia y Raquel,
a toda mi familia,
a mis amigos y compañeros
y a mi novia Patricia

Resumen

Este documento contiene la memoria del proyecto fin de carrera en el cual se ha realizado un sistema portátil capaz de capturar los diversos movimientos realizados por una persona para su futuro análisis.

Con este objetivo, se ha realizado un estudio previo del estado actual de las tecnologías necesarias para la composición de la aplicación. A partir de los conocimientos adquiridos, se ha diseñado un programa informático que cumple con los requisitos establecidos en el planteamiento inicial.

Finalmente, se ha implementado el sistema en un dispositivo concreto y evaluado detalladamente para la comprobación del resultado obtenido.

Índice

1. Introducción.....	1
1.1 Motivación del Proyecto.....	1
1.2 Ámbito del Proyecto.....	2
1.3 Objetivos del Proyecto	2
1.4 Organización de la Memoria	3
2. Estado del Arte	5
2.1 Dispositivos Móviles	5
2.1.1 Definición de Dispositivo Móvil	6
2.1.2 Características de los Dispositivos Móviles	7
2.1.3 Clasificación de los Dispositivos Móviles	8
2.1.4 Aplicaciones con Dispositivos Móviles	9
2.1.5 Palm Tungsten E2.....	10
2.2 Sensores de Aceleración.....	12
2.2.1 Definición de Sensor de Aceleración	12
2.2.2 Modelo Sensor de Aceleración de la Aplicación.....	13
2.3 Sensores RFID	14
2.3.1 Definición de Sensor RFID	14
2.3.2 Modelo Sensor RFID de la Aplicación	14
2.4 Bluetooth	15

2.4.1	Definición de Bluetooth	15
2.4.2	Historia de Bluetooth.....	16
2.4.3	Características de Bluetooth.....	17
2.4.4	Arquitectura de Protocolos Bluetooth	18
2.4.5	Perfiles Bluetooth	20
2.4.6	Conexiones Bluetooth	21
2.4.7	Dispositivos Bluetooth	23
2.5	Programación en Dispositivos Móviles.....	25
2.5.1	Limitaciones Hardware.....	25
2.5.2	Limitaciones Software	28
2.5.3	Sistemas Operativos	29
2.5.4	Lenguajes de Programación.....	30
2.6	Programación con Bluetooth.....	30
2.6.1	Programación mediante Puerto Serie	31
2.6.2	Programación mediante Sockets Bluetooth.....	32
2.6.3	API Java para Bluetooth	33
2.7	Programación para Tungsten E2	33
2.7.1	Puerto Serie de Tungsten E2	34
2.7.2	Sockets Bluetooth de Tungsten E2.....	35
3.	Gestión del Proyecto.....	36
3.1	Descomposición en Tareas.....	36
3.1.1	Actividad A: Análisis del Problema	36
3.1.2	Actividad B: Documentación y Análisis del Estado del Arte	37
3.1.3	Actividad C: Diseño de la Aplicación	38
3.1.4	Actividad D: Implementación de la Aplicación	39
3.1.5	Actividad E: Evaluación de la Aplicación.....	39

3.1.6	Actividad F: Redacción de la Memoria.....	40
3.2	Duración	40
3.3	Diagrama de Gannt	41
3.4	Presupuesto.....	44
3.4.1	Costes de Personal	44
3.4.2	Costes de Equipamiento	44
3.4.3	Coste Total.....	45
4.	Memoria del Trabajo Realizado	46
4.1	Consideraciones Previas al Diseño de la Aplicación	46
4.2	Fases de la Aplicación	47
4.2.1	Menú.....	48
4.2.2	Configuración	49
4.2.3	Conexión con los Dispositivos	49
4.2.4	Captura de Datos	49
4.2.5	Muestra de Resultados.....	50
4.2.6	Transferencia de Datos	50
4.2.7	Borrado de Datos	50
4.2.8	Manual de Usuario	50
4.3	Diseño de la Aplicación.....	50
4.3.1	Formato de la Aplicación	51
4.3.2	Modelo de Diseño en Tres Capas	51
4.3.3	Capa de Presentación.....	53
4.3.4	Capa de Negocio.....	55
4.3.5	Capa de Datos	65
4.3.6	Interfaz.....	66
4.3.7	Base de Datos	68

4.3.8	Ficheros de Salida.....	70
4.4	Implementación de la Aplicación.....	71
4.4.1	Configuración del Desarrollo	71
4.4.2	Interfaz de Usuario	72
4.4.3	Código Fuente	76
4.4.4	Descripción del Funcionamiento de la Aplicación.....	77
5.	Evaluación de la Aplicación	80
5.1	Definición Pruebas de Evaluación	80
5.2	Definición Tablas de Resultados de Evaluación.....	82
5.3	Realización Pruebas de Evaluación	85
5.3.1	Pruebas de Evaluación del Funcionamiento.....	85
5.3.2	Pruebas de Evaluación de la Calidad.....	91
5.3.3	Pruebas de Evaluación del Rendimiento	96
5.4	Resultados de la Evaluación.....	112
6.	Conclusiones.....	116
7.	Trabajos futuros	119
	Anexos.....	121
A	Manual de Usuario	121
A.1	Instalación del Ejecutable.....	121
A.2	Ejecución de la Aplicación	122
A.3	Consejos para una Correcta Ejecución	130
B	Configuración de Dispositivos Bluetooth	131
C	Ejemplo Conexión Bluetooth mediante Puertos COM	137
C.1	Funcionamiento de la Aplicación	137
C.2	Ejecución de la Aplicación	137

D	Ejemplo Práctico de Uso del API de Java para Bluetooth.....	140
D.1	Funcionamiento de la Aplicación	140
D.2	Ejecución de la Aplicación	140
E	Ejemplo de Conexión mediante el Puerto Serie de Palm.....	142
E.1	Funcionamiento de la Aplicación	142
E.2	Ejecución de la Aplicación	142
F	Ejemplo Conexión mediante Sockets Bluetooth de Palm.....	144
F.1	Funcionamiento de la Aplicación	144
F.2	Ejecución de la Aplicación	145
G	Contenido del CD	147
Referencias		148
Acrónimos		152

Índice de Figuras

Figura 1: Palm Tungsten E2	10
Figura 2: Sensor de aceleración WiTilt v2.5	13
Figura 3: Pila de protocolos Bluetooth	18
Figura 4: Conjunto de perfiles Bluetooth	21
Figura 5: Ejemplos <i>piconets</i> y <i>scatternet</i>	22
Figura 6: Dispositivos Bluetooth.....	23
Figura 7: Listado de tareas del proyecto.....	42
Figura 8: Diagrama de Gantt	43
Figura 9: Diagrama de casos de uso	47
Figura 10: Icono de la aplicación	51
Figura 11: Modelo de programación tres capas	52
Figura 12: Diagrama de comunicación entre capas.....	53
Figura 13: Diagrama de clases Capa Presentación.....	54
Figura 14: Diagrama de clases Capa de Negocio	55
Figura 15: Interfaz InterfazNegocioPresentacion.....	56
Figura 16: Clase GestorPantalla	56
Figura 17: Clase GestorEventos	57
Figura 18: Clase GestorConexiones	58
Figura 19: Clase GestorBluetooth	60
Figura 20: Clase GestorDatos.....	62
Figura 21: Clase GestorEnvio	63
Figura 22: Clase GestorParametros	64
Figura 23: Diagrama de clases Capa Datos	65
Figura 24: Diagrama de navegación de formularios	67
Figura 25: Creación de ficheros de transferencia	70
Figura 26: Ejemplo de fichero de transferencia.....	71
Figura 27: Formulario de bienvenida	73

Figura 28: Formulario Menú	73
Figura 29: Formulario Configuración	74
Figura 30: Formulario Ejecución.....	74
Figura 31: Formulario Resultado.....	75
Figura 32: Formulario Transferencia.....	75
Figura 33: Formulario Borrado	76
Figura 34: Formulario Ayuda.....	76
Figura 35: Porción de código del fichero creado en la evaluación de transferencia	89
Figura 36: Número de datos erróneos según frecuencia del sensor	93
Figura 37: Número de datos erróneos según frecuencia de configuración.....	93
Figura 38: Porcentaje de datos por cada sensor.....	96
Figura 39: Comparativa del rendimiento de los sensores de aceleración con el PC	99
Figura 40: Comparativa del rendimiento de los sensores de aceleración con la PDA .	101
Figura 41: Variaciones en distintas ejecuciones con un sensor de aceleración.....	102
Figura 42: Comparativa frecuencias según número de sensores.....	113
Figura 43: Comparativa frecuencias según número de sensores.....	114
Figura 44: Árbol de directorios y ficheros.....	122
Figura 45: Pantalla de Bienvenida.....	123
Figura 46: Pantalla Menú	123
Figura 47: Pantalla de Configuración.....	124
Figura 48: Pantalla de Conexión	124
Figura 49: Pantalla de Listado de Dispositivos	125
Figura 50: Pantalla de Conexiones Establecidas	125
Figura 51: Pantalla de Muestra de Datos.....	125
Figura 52: Pantalla de Resultados	126
Figura 53: Pantalla de Transferencia por Bluetooth.....	126
Figura 54: Pantalla de Listado de Dispositivos para la Transferencia	127
Figura 55: Pantalla de Transferencia	127
Figura 56: Pantalla de Finalización de Transferencia	128
Figura 57: Pantalla de Confirmación de Borrado de Datos.....	128
Figura 58: Pantalla de Finalización de Borrado de Datos	129
Figura 59: Pantallas de Ayuda.....	129
Figura 60: Pantalla de Despedida	130
Figura 61: Ventana Agregar Dispositivo Bluetooth.....	131
Figura 62: Ventana Seleccionar Dispositivo Bluetooth	132

Figura 63: Ventana Clave de Paso Dispositivo Bluetooth	132
Figura 64: Ventana Pasos Conexión.....	133
Figura 65: Ventana Puertos COM	133
Figura 66: Ventana Conexión Terminal	134
Figura 67: Ventana Puerto Terminal	134
Figura 68: Ventanas Configuración Parámetros Terminal	135
Figura 69: Ventana Datos Terminal	135
Figura 70: Dispositivo GPS	138
Figura 71: Ejecución ejemplo conexión Bluetooth mediante puertos COM.....	138
Figura 72: Ejemplo API Java para Bluetooth.....	141
Figura 73: Ejemplo conexión puerto serie Palm	143
Figura 74: Mando videoconsola Wii	144
Figura 75: Ejemplo conexión mediante sockets Bluetooth	145
Figura 76: Organización del CD.....	147

Índice de Tablas

Tabla 1: Características Tungsten E2	11
Tabla 2: Conexiones Tungsten E2.....	11
Tabla 3: Comparativa tecnologías inalámbricas.....	18
Tabla 4: Valores de Tipo de Dispositivo Bluetooth	24
Tabla 5: Dimensiones de dispositivos móviles	26
Tabla 6: Duración de las tareas del proyecto.....	41
Tabla 7: Costes de personal	44
Tabla 8: Costes de equipamiento.....	45
Tabla 9: Definición de campos de los registros de la base de datos.....	69
Tabla 10: Ejemplo registro de lectura a un sensor de aceleración.....	69
Tabla 11: Ejemplo registro de lectura a un sensor RFID	69
Tabla 12: Ficheros con código fuente de la aplicación	77
Tabla 13: Plantilla de test de funcionamiento de conexiones.....	82
Tabla 14: Plantilla de test de funcionamiento de transferencia	83
Tabla 15: Plantilla de test de calidad	84
Tabla 16: Plantilla de test de rendimiento	85
Tabla 17: Test de funcionamiento con un sensor de aceleración	86
Tabla 18: Test de funcionamiento con un sensor RFID	86
Tabla 19: Test de funcionamiento con tres sensores de aceleración	87
Tabla 20: Test de funcionamiento con tres sensores de aceleración y uno RFID.....	88
Tabla 21: Test de funcionamiento transferencia de una ejecución.....	88
Tabla 22: Test de funcionamiento transferencia de tres ejecuciones	90
Tabla 23: Test de funcionamiento transferencia tras apagado	91
Tabla 24: Test de calidad con un sensor de aceleración.....	92
Tabla 25: Test de calidad con un sensor RFID.....	94
Tabla 26: Test de calidad con varios sensores.....	95
Tabla 27: Test de rendimiento de los sensores con el ordenador	98

Tabla 28: Test de rendimiento con conexión al sensor 00:A0:96:1B:30:95	100
Tabla 29: Test de rendimiento con conexión a un sensor 00:A0:96:18:C6:03	100
Tabla 30: Test de rendimiento con conexión a un sensor 00:A0:96:1B:30:D0.....	100
Tabla 31: Test de rendimiento con conexión a un sensor 00:A0:96:17:A4:CD.....	100
Tabla 32: Test de rendimiento con conexión a un sensor 00:A0:96:1B:32:C6.....	101
Tabla 33: Test de la variación del rendimiento del sensor 00:A0:96:1B:30:95	102
Tabla 34: Test de rendimiento con conexión a un sensor de aceleración.....	103
Tabla 35: Test de rendimiento con conexión con dos sensores de aceleración.....	104
Tabla 36: Test de rendimiento de comprobación del retardo por mostrar los datos	105
Tabla 37: Test de rendimiento con conexión con tres sensores de aceleración	105
Tabla 38: Test de rendimiento con conexión con cuatro sensores de aceleración	106
Tabla 39: Test de rendimiento con conexión con cinco sensores de aceleración.....	107
Tabla 40: Test de rendimiento con conexión con un sensor RFID	107
Tabla 41: Test de rendimiento con conexión al sensor RFID y uno de aceleración	108
Tabla 42: Test de rendimiento con conexión al sensor RFID y cinco de aceleración..	109
Tabla 43: Ejemplos con su ocupación de memoria	110
Tabla 44: Estimación de la ocupación de memoria.....	111
Tabla 45: Análisis de la duración de las baterías.....	112
Tabla 46: Desglose de lecturas del test de rendimiento con todos los sensores.....	114
Tabla 47: Información acerca del mando de la videoconsola Wii	145

1. Introducción

En este capítulo se resume la información preliminar que el lector debe adquirir para la total comprensión de este documento. En primer lugar, se introduce el proyecto propiamente dicho a través de su motivación, su ámbito y características y sus principales objetivos. Posteriormente, se expone la manera en que ha sido estructurado este documento enumerando cada uno de los capítulos e incluyendo una breve descripción de los mismos.

1.1 Motivación del Proyecto

Por su naturaleza los seres humanos pueden realizar una serie de movimientos corporales ordenados por el cerebro y que, en ocasiones, pueden responder a estímulos externos recibidos a través de los sentidos.

En la actualidad existen dispositivos destinados a medir estos desplazamientos, es decir, son capaces de detectar las variaciones de posición en los tres ejes de coordenadas, así como la relación de estas modificaciones con el tiempo. Por tanto, mediante la colocación de estos aparatos en las extremidades, los movimientos pueden ser registrados numéricamente y posteriormente analizados. Además, a día de hoy se dispone de dispositivos capaces de identificar objetos para examinar los elementos del entorno que el usuario utiliza.

A partir de todo lo anterior, se puede conformar un sistema de estudio de los movimientos que son llevados a cabo por las personas en un escenario determinado, obteniendo tanto los desplazamientos realizados como los objetos utilizados. Este sistema debe tener un núcleo central capaz de recibir toda la información recogida por los diferentes sensores. Para ello, se necesita una unidad de procesamiento como puede ser un ordenador.

Sin embargo, si se desea que el sistema sea transportable y robusto frente a escenarios exteriores es ideal que el computador sea un dispositivo móvil. En estos tiempos, se dispone de una gran variedad de potentes terminales móviles que cumplen los requisitos indicados de funcionalidad y portabilidad como son las PDA.

Otra cuestión relevante en la definición de la aplicación consiste en la comunicación entre los distintos dispositivos, que son incluidos en el sistema para la

recepción de los movimientos, y la unidad central. Las emergentes tecnologías de redes inalámbricas suponen un eficiente recurso para el sistema y, en concreto, Bluetooth reúne las condiciones de conexión y transferencia deseadas.

1.2 Ámbito del Proyecto

El sistema anteriormente introducido se nos presenta con la intención de realizar un análisis de los movimientos realizados por un ser humano en un escenario determinado. En concreto, se desea realizar un sistema capaz de reconocer las actividades que realiza una persona a partir de los datos recogidos por unos sensores. La información obtenida del mismo puede ser utilizada siguiendo dos patrones de evaluación.

El primero de ellos consistiría en estudiar diferentes ejecuciones de las mismas actividades por parte de varias personas con el fin de obtener un conjunto de movimientos óptimos a la hora de lograr un objetivo.

Mediante el segundo, se podría obtener una comparación entre una serie de pasos ideales prefijados con una realización concreta de los movimientos.

Es decir, el sistema puede ser empleado tanto para la creación de un patrón de movimientos a partir de pruebas empíricas, como para la evaluación de un ejemplo concreto con un modelo de comportamiento prefijado como el óptimo. A modo de ejemplo, una aplicación que puede tener un sistema de estas características se fundamenta en el campo de la medicina y más concretamente en el de la fisioterapia. Para la rehabilitación de las diferentes articulaciones, los pacientes de hospitales o clínicas ejecutan movimientos periódicos que deben ser realizados acercándose lo más posible a un patrón.

Otros campos tecnológicos en los que puede aplicarse este estudio son la robótica, para la creación de cuerpos que realicen movimientos automatizados o la realidad virtual para la representación en un mundo informático de desplazamientos humanos.

1.3 Objetivos del Proyecto

Partiendo de la introducción del proyecto realizada en los apartados previos, se pueden definir una serie de objetivos que tendrá que cumplir el desarrollo de este proyecto.

El primero y más general consiste en desarrollar un sistema capaz de registrar los movimientos de un cuerpo en un escenario móvil con dispositivos inalámbricos. Como se ha comentado anteriormente, con esta finalidad se van a emplear una serie de sensores que generarán datos y que a su vez deberán ser recogidos para su futuro análisis utilizando un dispositivo móvil determinado. Tanto los modelos de los sensores

como el dispositivo móvil han sido impuestos y proporcionados por el tutor del proyecto y, por tanto, su utilización son un requisito indispensable.

Para conseguir este objetivo se han establecido los siguientes subobjetivos:

- Estudio de la situación actual de los diferentes dispositivos inalámbricos existentes en el mercado capaces de ser el núcleo central de esta aplicación.
- Análisis de la naturaleza y el funcionamiento de los sensores que se van a emplear en el sistema.
- Investigación acerca de las diferentes tecnologías de comunicación inalámbrica y, en profundidad, de las capacidades y posibilidades de la tecnología Bluetooth.
- Diseño e implementación de un programa informático que permita el desarrollo del sistema deseado y genere los ficheros con los datos recogidos.
- Evaluación de las capacidades de la aplicación y comprobación del cumplimiento de los objetivos planteados.

1.4 Organización de la Memoria

El presente documento describe la realización del proyecto fin de carrera y está estructurado en los siguientes apartados:

- **Apartado 1: Introducción**

Descripción general del proyecto y exposición de la estructura de la memoria.

- **Apartado 2: Estado del arte**

Presentación de los dispositivos y las tecnologías que han sido utilizadas en el proyecto. Se analiza su estado actual, tanto sus características como sus aplicaciones. Se describen los dispositivos móviles, los sensores de aceleración, los sensores RFID y la tecnología inalámbrica Bluetooth. Además, se resumen las posibilidades de programación existentes para los dispositivos móviles con conexión Bluetooth.

- **Apartado 3: Gestión del proyecto**

En este capítulo se muestra la metodología empleada para llevar a cabo la realización del proyecto. En él se exponen las tareas en las que se ha dividido así como la duración de cada una y el consiguiente gasto.

- **Apartado 4: Objetivos del proyecto**

Resumen de la finalidad del proyecto y, por tanto, los objetivos que se querían alcanzar en su realización.

- **Apartado 5: Memoria del trabajo realizado**

En esta parte se presenta y describe el trabajo realizado en este proyecto fin de carrera. En primer lugar, se introducen varios requisitos que la aplicación debe cumplir. A continuación, se muestran las fases en las que se divide el funcionamiento de la aplicación. Seguidamente, se realiza el diseño completo del programa. Para concluir, se documenta la implementación final de la aplicación.

- **Apartado 6: Evaluación de la aplicación**

Presentación de las principales pruebas que se realizan para la evaluación de la aplicación así como la recogida y el análisis de los resultados obtenidos en las mismas.

- **Apartado 7: Conclusiones**

Redacción de las principales conclusiones finales como consecuencia del trabajo realizado en este proyecto.

- **Apartado 8: Trabajos futuros**

Resumen de las principales ampliaciones posibles para llevar a cabo como continuación o alternativa a este proyecto.

- **Apartado 9: Anexos**

Información adicional que se incluye de manera auxiliar para completar este documento.

2. Estado del Arte

En este apartado se redacta el análisis inicial realizado acerca del estado en que se encuentran en la actualidad las tecnologías utilizadas en este proyecto. Este estudio ha sido desarrollado previamente al diseño de la aplicación y ha servido de guía introductoria y referencia para las siguientes fases del proyecto.

La metodología empleada ha consistido en una primera etapa de investigación puramente teórica en la que se han identificado términos, agrupado conceptos y relacionado temas. Mediante este estudio, se han reforzado las nociones previamente adquiridas acerca de algunos conceptos y se han aprendido otros completamente desconocidos en un principio.

Posteriormente, el estudio se ha centrado en una perspectiva más práctica. Se ha sondeado el mercado para enumerar los dispositivos existentes en la actualidad. Además, se han buscado, estudiado y evaluado las aplicaciones en las que se implementan dichas tecnologías.

La documentación de esta sección ha sido dividida en base a los diferentes conceptos sobre los que se ha realizado el estudio debido a que su uso es un requisito indispensable en este proyecto.

2.1 Dispositivos Móviles

Como se comentó en el apartado introductorio de este documento, el módulo central del sistema que se debe crear en este proyecto consiste en un dispositivo móvil, el cual debe tener un conjunto de características técnicas para satisfacer una serie de requisitos mínimos de procesamiento, memoria y conectividad. En este apartado, se va a analizar el estado actual de los dispositivos móviles, sus características, sus funcionalidades y los modelos existentes.

2.1.1 Definición de Dispositivo Móvil

No existe una definición formal de estos aparatos. Coloquialmente, se describe a los dispositivos móviles, también conocidos por el término inglés *handhelds*, como unas computadoras tan pequeñas que entran en un bolsillo. De una manera más formal, se definen como aparatos con las siguientes características:

- Tamaño reducido
- Ciertas capacidades de procesamiento
- Móviles
- Con conexión permanente o intermitente a una red
- Memoria limitada

Dependiendo del nivel de funcionalidad se clasifica a los dispositivos móviles en tres categorías:

- **Dispositivo móvil de datos limitados:** tienen una pequeña pantalla de tipo texto con servicios de datos limitados. Un ejemplo serían los teléfonos móviles sencillos.
- **Dispositivo móvil de datos básicos:** tienen una pantalla de mediano tamaño (entre 120 x 120 y 240 x 240 píxeles) así como una navegación basada en íconos por medio de una rueda o cursor. Ofrecen acceso a correo electrónico y navegador web. Dentro de este tipo se encuadran los teléfonos móviles inteligentes o *smartphones*.
- **Dispositivo móvil de datos mejorados:** tienen pantallas de medianas a grandes (por encima de los 240 x 240 píxeles), navegación táctil y funcionalidad similar a la expuesta en la anterior categoría pero con más aplicaciones nativas en versión móvil. En este tipo se incluyen, por ejemplo, los Asistentes Electrónicos Personales, más conocidos como las PDA.

Debido al auge en el desarrollo de estos dispositivos y a la cantidad de modelos existentes en el mercado, las distancias entre cada una de las anteriores categorías se han reducido y supone un esfuerzo mayor categorizar a cada uno de ellos.

Para los siguientes apartados, el estudio se centrará en las dos últimas categorías y, por tanto, las más completas, descartando los dispositivos móviles de datos limitados ya que poseen una funcionalidad demasiado baja para el desarrollo de este sistema.

2.1.2 Características de los Dispositivos Móviles

Actualmente, un dispositivo móvil tiene el conjunto de características descrito a continuación.

- **Pantalla táctil:** sirve para interactuar con el usuario de manera cómoda, rápida y sencilla. Los terminales pueden incluir un lápiz desmontable que ayuda en la selección de elementos. Comúnmente, para agregar texto puede utilizarse uno de los siguientes métodos:
 - Teclado virtual: se muestra en la pantalla la imagen de un teclado compuesto por teclas que pueden ser pulsadas y que representan un carácter cada una.
 - Teclado externo, similar al de un computador, que se conecta vía USB o Bluetooth.
 - Sistema de reconocimiento de escritura de letras o palabras. Consiste en un software que lee los trazos realizados sobre un sector de la pantalla y los traduce en letras en base al reconocimiento de patrones de las formas de grafía de las letras en la escritura sobre papel.
 - Reconocimiento de símbolos. Sistema similar al anterior en el que un cierto grupo de símbolos representan una letra.

Por otro lado existen equipos con teclados completos y teclas de desplazamiento que imitan el ingreso de información típico de los ordenadores.

En la actualidad, los modelos de dispositivos móviles más modernos incluyen una nueva técnica de interacción basada en el reconocimiento simultáneo de múltiples puntos de contacto.

- **Ranuras para tarjetas de memoria:** en nuestros días se encuentra muy extendido el uso de tarjetas externas de memoria. Los modelos modernos de dispositivos móviles incluyen una ranura para insertar estas tarjetas y aumentar así las capacidades de almacenamiento. Además, permiten la transferencia directa de archivos sin necesidad de programas intermediarios.
- **Conectividad por cable:** una funcionalidad típica de estos dispositivos es la de sincronizar datos con un equipo de sobremesa u ordenador portátil. Para ello, existen dos tipos de conexiones por cable. Por un lado existe la conexión por cable serial, es decir, conectando por el puerto serie. Por otro lado, los dispositivos más modernos usan un cable USB. Al mismo tiempo que permiten la conexión con el ordenador, sirven como puertos de alimentación de corriente eléctrica.

- **Conectividad inalámbrica:** muchos de los dispositivos móviles tienen diferentes métodos de establecer conexiones inalámbricas. El más extendido es la conectividad Bluetooth, permitiendo conectar teclados externos, auriculares, manos libres, GPS y más accesorios en distancias de pocos metros. Otro método existente aunque menos utilizado y con menos rendimiento es la conexión mediante infrarrojos. Además, los dispositivos más actuales incluyen conectividad Wi-Fi para permitir la conexión inalámbrica a Internet.
- **Instalación de software:** una característica importante, de prácticamente la totalidad de los dispositivos móviles, es la posibilidad de instalación de programas para incrementar el procesamiento de datos y la conectividad. Normalmente estas aplicaciones pueden ser desarrolladas tanto por el fabricante, como por el operador como por un tercero.

2.1.3 Clasificación de los Dispositivos Móviles

Como se comentó anteriormente, existen dos grandes tipos de dispositivos portátiles cuyo uso se encuentra enormemente extendido y cuyo desarrollo se ha incrementado. Estos son las PDA y los teléfonos inteligentes. El origen de ambos es muy distinto, pero en la actualidad tanto su apariencia como su funcionalidad cada vez son más similares.

Una PDA es un ordenador de mano originalmente diseñado como agenda electrónica que hoy en día se puede utilizar como una computadora doméstica. Dentro de las PDA se tienen dos grandes familias de dispositivos: las Palm y las PocketPC. Las Palm son las PDA que utilizan el sistema operativo PalmOS creado por la empresa PalmSource, la cual fue absorbida en el año 2005 por la compañía japonesa Access Co. La versión 5 de este sistema operativo también es denominada Garnet OS. Por el contrario, las PocketPC ejecutan un sistema operativo de la familia Windows, concretamente el Windows CE o Windows Mobile. Incluyen versiones reducidas de aplicaciones típicas de la versión de este sistema operativo para ordenadores como Microsoft Outlook, Internet Explorer o Microsoft Word. También existen dispositivos de la familia Linux pero se hallan menos extendidos.

El otro gran grupo de dispositivos móviles son los denominados teléfonos inteligentes. Funcionan como un teléfono celular pero con características similares a un ordenador personal. En la actualidad, el desarrollo de estos dispositivos está alcanzando cuotas de mercado muy elevadas e incluyen las tecnologías más avanzadas. Un ejemplo claro de estos dispositivos es el denominado Blackberry, cuya principal característica es la de permitir el acceso al correo electrónico de manera segura.

Además de estos dos grandes grupos, existen multitud de aparatos por cuyas características pueden ser considerados dispositivos móviles. Algunos ejemplos pueden ser consolas portátiles, terminales, buscapersonas o sistemas de navegación de automóviles.

Actualmente, el mercado de los dispositivos móviles continúa en plena expansión. El número de terminales vendidos sigue en aumento y con el desarrollo de nuevos materiales y tecnologías las características y funcionalidades de los aparatos evolucionan a gran velocidad.

En cuanto a las PDA, las marcas comerciales más destacadas de venta de dispositivos son Acer, Asus, Casio, Compaq, Dell, Fujitsu-Siemens, Handspring, HP, Palm, PSion, Sharp, Sony y Toshiba. Sus características y diseños son muy variados.

Sin embargo, los terminales que más auge están alcanzando en la segunda mitad de esta década son los teléfonos móviles inteligentes. La irrupción de los modelos más modernos como el Iphone de Apple, el OMNIA de Samsung, el G1 de T-Mobile o el N96 de Nokia, han proporcionado un nuevo impulso tanto en ventas como en funcionalidad.

2.1.4 Aplicaciones con Dispositivos Móviles

Los dispositivos móviles y su amplia implantación entre la sociedad han provocado su rápido desarrollo y el de las posibilidades que ofrecen a sus usuarios. En estos últimos años se están convirtiendo en una auténtica plataforma de ejecución de aplicaciones informáticas. Los consumidores pueden hacer uso de una serie de productos y servicios que facilitan la realización de su trabajo y pueden mejorar el aprovechamiento de su tiempo libre.

Según su utilidad, sus aplicaciones se pueden agrupar en los dos siguientes grupos:

- **Usos Generales:** de entre los que se destacan los siguientes tipos de ejemplos y herramientas.
 - **Localización:** localización de personas en tiempo real, callejeros y guías de ciudades, estado del tráfico.
 - **Comercio electrónico:** compra de entradas.
 - **Correo electrónico:** consulta y envío de correos electrónicos.
 - **Entretenimiento y ocio:** videojuegos, chats, descarga de música y vídeos.
 - **Navegación:** acceso a redes y obtención de información.
- **Usos Para las Empresas:** las principales aplicaciones de este grupo son enumeradas a continuación.
 - **Bases de datos:** los comerciales podrán consultar datos sobre sus clientes o productos desde cualquier punto.

- **Correo electrónico:** conexión entre el personal de la empresa mediante el correo electrónico.
- **Transferencia de ficheros:** uso de los terminales para realizar la transferencia de ficheros.
- **Acceso a Internet y a Intranet:** las empresas facilitan los accesos desde dispositivos móviles.

2.1.5 Palm Tungsten E2

A continuación, se realiza una descripción del modelo de dispositivo portátil que ha sido empleado en el desarrollo de este proyecto. Uno de los requisitos del sistema es el de la utilización de la PDA Palm Tungsten E2 [2], cuya imagen se puede observar en la siguiente figura.



Figura 1: Palm Tungsten E2

El modelo de PDA Tungsten E2 fue introducido el 13 de abril de 2005 reemplazando a su antecesor y de similar nombre Tungsten E [3].

Pertenece a la serie Tungsten, la cual fue la línea de negocio de la empresa Palm, Inc hasta la compra de la firma Palm. Desde entonces, se ha eliminado el nombre Tungsten de sus nuevos productos. De este modo, en 2007 el modelo Tungsten E2 era el único que continuaba usando el término. El resto de productos desarrollados posteriormente pese a prescindir del nombre continúan con la misma línea.

Su precio de lanzamiento fue 249 dólares. En octubre de 2005, su coste se había reducido a 199 dólares. En la actualidad en la página oficial de Palm España se puede comprar por 199 euros.

El contenido software incluye programas típicos en estos dispositivos como Favoritos, Calendario, Contactos, Tareas, Notas o Calculadora.

Las características técnicas del dispositivo están resumidas en la siguiente tabla [4]:

Característica	Descripción
Sistema Operativo	Palm OS® Garnet (v5.4) MultiIdioma
Procesador	Intel Xscale a 200MHz
Memoria	32 MB de RAM (capacidad de almacenamiento real de 26 MB). 8 MB de ROM
Batería	De iones de litio recargable con una duración de 8 días
Pantalla	TFT transfect/reflectiva con retroiluminación de 320x320 de alta resolución y 65.536 colores
Ranura de expansión	Admite tarjetas de expansión SD, SDIO y MultiMediaCard. El tamaño máximo de tarjeta SD soportada es de 1GB debido a limitaciones de hardware
Botones	Botón navegador de 5 direcciones.
Conexión para auriculares estéreo	Admite auriculares estéreo estándar para la reproducción de audio
Sonido	Digital (MP3)
Altavoz	En la parte posterior
Lápiz óptico	Incorporado
Tamaño	11,4 x 7,8 x 1,5 cm
Peso	133 g

Tabla 1: Características Tungsten E2

Por otro lado, las posibilidades de conexión son las siguientes:

Conexión	Descripción
Puerto MultiConector	Para sincronización con el PC (USB)
Bluetooth	Bluetooth 1.1
Puerto Infrarrojos	IrDA
Wi-Fi	Se puede insertar una tarjeta SDIO Wi-Fi fabricada por Palm

Tabla 2: Conexiones Tungsten E2

Asimismo, cabe destacar que existen en el mercado multitud de accesorios para aumentar las capacidades de este dispositivo como antenas GPS, tarjetas de red, teclados inalámbricos, o tarjetas de expansión.

2.2 Sensores de Aceleración

Como se comenta en el apartado de introducción, para el desarrollo de la aplicación se van a utilizar unos sensores de aceleración. Estos serán distribuidos por distintas posiciones en el cuerpo del usuario y proporcionarán los datos de posición que registrarán los movimientos realizados por la persona a estudiar.

2.2.1 Definición de Sensor de Aceleración

Como su propio nombre indica, un sensor de aceleración, también conocido como acelerómetro, es un instrumento destinado a medir aceleraciones [5]. Utiliza el principio físico de la Ley Fundamental de la Dinámica o Segunda Ley de Newton basada en la fórmula: $F = ma$, donde F representa las fuerzas que actúan sobre la masa m y a es la aceleración del movimiento.

Tradicionalmente, se construye uniendo una masa a un dinamómetro cuyo eje está en la misma dirección que la aceleración. Dado que el dinamómetro indica el módulo de F y el valor de m es una característica del instrumento que no varía, se puede conocer el módulo de la aceleración a .

Según la dirección en la que miden la aceleración, los acelerómetros pueden ser uniaxiales, biaxiales o triaxiales, obteniendo datos de uno, dos y tres ejes respectivamente.

Actualmente, su uso se encuentra muy extendido en el campo de la mecánica de automoción para calcular las aceleraciones laterales a las que se ve sometido un cuerpo al tomar una curva, la deceleración que sufre un cuerpo humano durante un accidente o en ensayos de frenado.

Por otro lado, en el campo de la informática y más concretamente el de la robótica, estos dispositivos se utilizan para medir el movimiento y las vibraciones a las que está sometido un robot, o una parte de él, y la inclinación con respecto a la gravedad.

2.2.2 Modelo Sensor de Aceleración de la Aplicación

El modelo de acelerómetro concreto elegido para utilizar en este sistema es el WiTilt v2.5 fabricado por la compañía SparkFun [7]. Como se puede ver en la Figura 2, su tamaño es reducido (5.6 x 7.1 x 1.9 centímetros), así como su peso (57 gramos), y por tanto, es manejable para el transporte y desplazamiento en las ejecuciones que se quieren realizar.



Figura 2: Sensor de aceleración WiTilt v2.5

Sus principales características son que emplea una detección en los tres ejes X, Y, Z, que su rango y calibración es ajustable desde un menú interactivo y que posee conexión Bluetooth para una robusta conexión con una gran variedad de aparatos.

La información proporcionada por el dispositivo se envía por Bluetooth codificada en cadenas de un máximo de veintiocho caracteres que siguen el formato siguiente:

$$X = V_X \text{ SP } Y = V_Y \text{ SP } Z = V_Z \text{ SL RC}$$

Los valores de las coordenadas V_X , V_Y y V_Z son números de como máximo seis caracteres como por ejemplo los siguientes dos:

$$1.042 \text{ ('1', '.', '0', '4', '2')}$$

$$-0.381 \text{ ('-', '0', '.', '3', '8', '1')}.$$

Los símbolos SP, SL y RC representan un espacio (' '), un salto de línea ('\n') y un retorno de carro ('\r') respectivamente.

Por todo esto, una cadena de ejemplo podría ser la siguiente:

$$X = 1.042 \text{ Y}=-0.381 \text{ Z}=572$$

En la actualidad existe una versión más actualizada del acelerómetro, el WiTilt v3.0 [8], el cual incluye, entre otras mejoras, una batería recargable y una carcasa de protección.

En Internet se pueden encontrar varias aplicaciones que emplean estos dispositivos para recoger datos de aceleración o inclinación [9] [10]. En ellas, se muestran gráficas de cómo varían los datos recibidos del sensor y posteriormente analizados. Por otro lado, existe una aplicación que registra y analiza los datos recogidos por un sensor de este modelo en un salto al vacío de un paracaidista [11].

2.3 Sensores RFID

Conjuntamente con los sensores de aceleración, en el sistema del proyecto se incluye el uso de sensores RFID con el fin de registrar los objetos que el usuario toque o sujete.

2.3.1 Definición de Sensor RFID

Los sensores RFID se basan en la tecnología del mismo nombre [12]. Estas siglas representan el término inglés *Radio Frequency IDentification*, cuyo significado en español es identificación por radiofrecuencia. Consiste en un sistema remoto de almacenamiento y recuperación de datos que usa dispositivos denominados etiquetas o transpondedores RFID. Su utilización se basa en la transmisión de la identidad de un objeto mediante ondas de radio. Las etiquetas RFID son unos dispositivos de reducido tamaño que se adhieren a un objeto. Contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Al emplear radiofrecuencia se evita la necesidad de visión directa entre emisor y receptor para su detección.

En la actualidad, las aplicaciones más conocidas de estos sensores son por ejemplo el seguimiento de libros en las bibliotecas, el control de acceso en edificios, la identificación de acreditaciones, el seguimiento de equipajes en aerolíneas, el seguimiento de artículos de ropa o la apertura de vehículos de alta gama.

2.3.2 Modelo Sensor RFID de la Aplicación

El modelo de dispositivo que se va a utilizar en el sistema es el SM3005 [13]. Sus principales características que lo hacen adaptable a la aplicación son su reducido tamaño, su facilidad de uso y su conexión mediante enlace RS232.

Para integrarlo en el sistema se han realizado dos ampliaciones. En primer lugar, se ha añadido una antena cuadrada de cobre que ha sido soldada al sensor [14]. Por otro lado, un dispositivo adaptador que convierte la comunicación de puerto serie a Bluetooth para poder recoger los datos de este dispositivo sin necesidad de cables de igual manera que con los sensores de aceleración [15].

La información proporcionada por el dispositivo se envía por Bluetooth y consiste en cadenas de caracteres que representan el identificador de las etiquetas que detecta. Se componen de diez números hexadecimales como por ejemplo 0105AA93BA.

En Internet se halla un ejemplo de programación aplicando este tipo de sensores. En él se desarrolla un juego basado en el reconocimiento de colores [16]. Para ello, se divide un tablero en colores y se añade una etiqueta RFID diferente a cada uno. El programa comprueba en cada momento que se selecciona, mediante contacto o acercamiento de una antena, el color indicado.

2.4 Bluetooth

Según la definición de este proyecto, la tecnología de comunicación entre los diferentes dispositivos que compondrán el sistema no debe ser cableada. De entre la variedad de sistemas inalámbricos que existen desarrollados en la actualidad, se escogió la comunicación Bluetooth como la encargada de establecer las conexiones. Esta elección fue realizada desde el mismo planteamiento del problema debido a sus características conocidas y a que los dispositivos de la aplicación están capacitados para su uso. En este apartado se va a definir qué es la tecnología Bluetooth así como sus principales características y usos.

2.4.1 Definición de Bluetooth

Con un gran número de dispositivos habilitados en la actualidad, incluyendo un alto porcentaje de los teléfonos móviles vendidos en el mundo, Bluetooth se presenta como la tecnología inalámbrica ideal para la conexión de dispositivos electrónicos.

Formalmente, Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura y globalmente libre.

Originalmente fue concebido como una alternativa inalámbrica a la transmisión de datos por cable mediante puerto serie. Posteriormente, se establecieron los siguientes objetivos con los que esta tecnología fue diseñada:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre dispositivos.

- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos.

Por su naturaleza, Bluetooth es un módulo de radio de baja potencia que puede integrarse en terminales de distinta naturaleza. En la actualidad, en los sectores de las telecomunicaciones y la informática personal, se incluye en aparatos como PDA, teléfonos móviles, ordenadores personales y portátiles, impresoras o cámaras digitales.

Además, esta tecnología permite la creación de redes de área personal conectando varios dispositivos mediante un enlace tanto de voz como de datos.

2.4.2 Historia de Bluetooth

En 1994 los ingenieros de la empresa sueca Ericsson iniciaron un estudio para investigar la viabilidad de una interfaz vía radio, de bajos costo y consumo, para la interconexión entre teléfonos móviles y otros accesorios con el objetivo de eliminar cables entre aparatos.

Cuatro años después, a principios de 1998 cuatro grandes compañías se unieron a Ericsson formando un Grupo de Especial Interés en Bluetooth (SIG) [20]. Estas fueron Nokia, IBM, Toshiba e Intel. El SIG es una organización privada y sin ánimo de lucro que no se ocupa de la fabricación y venta de productos con tecnología Bluetooth, sino de su desarrollo. La idea de este grupo era lograr un conjunto adecuado de áreas de negocio, ya que se hallaban en el grupo dos líderes del mercado de las telecomunicaciones, dos del mercado de los PC portátiles y un líder de la fabricación de chips.

La primera versión de la norma, 1.0A, se publicó en Julio de 1999, la cual ha sido modificada cinco veces, en 1.0, 1.1, 1.2, 2.0, 2.1 hasta la actualidad. A partir de la versión 1.2, se provee una solución inalámbrica complementaria para coexistir Bluetooth y Wi-Fi en el espectro de los 2.4 GHz sin interferencia entre ellos. La versión 2.0 fue creada para mejorar las velocidades de transmisión a la vez que intenta solucionar algunos errores en la anterior especificación. Por su parte, la versión 2.1 simplifica los pasos para crear la conexión entre dispositivos y reduce el consumo de potencia hasta cinco veces. Para mediados de 2009 está estipulado que salga a la luz una nueva versión, la 2.2, la cual aumenta considerablemente la velocidad de transferencia para acercar su trabajo al de Wi-Fi [21].

Actualmente, al SIG se han añadido miembros tales como Motorola, 3Com, Lucent y Microsoft, además de contar con el respaldo de 1900 empresas de tecnología y 2000 empleados de otras tantas empresas que investigan productos y servicios con aplicaciones Bluetooth. Está compuesto por más de 9.000 miembros, líderes en las áreas de telecomunicaciones, informática, industria automotriz, música, confección, automatización industrial y tecnología de redes.

2.4.3 Características de Bluetooth

Las características de la tecnología de comunicación Bluetooth más conocidas son su bajo consumo de energía, sus cortas distancias de alcance y el bajo precio de sus dispositivos.

Una frase que resume sus cualidades es la siguiente: especificación abierta que utiliza en lugar de cables un enlace radio de corto alcance para comunicar dispositivos de voz y datos de forma simple y barata, pudiendo ser utilizada en cualquier parte del mundo.

Sin embargo, de la página oficial del SIG, se obtiene una descripción más técnica de sus características mostradas a continuación:

- La tecnología inalámbrica Bluetooth está orientada a aplicaciones de voz y datos.
- Bluetooth funciona en la banda de frecuencia de 2.4 GHz, que no precisa de ninguna licencia.
- Bluetooth tiene un radio de acción de 10 o 100 metros dependiendo de la clase del dispositivo Bluetooth. La máxima velocidad de transmisión es de 3 Mbps.
- Los objetos sólidos no suponen ningún obstáculo para la tecnología inalámbrica Bluetooth.
- Con Bluetooth tampoco es necesario que los dispositivos estén situados en la misma línea de visión, es decir, orientados uno frente a otro, ya que se transmite en todas direcciones.
- La seguridad siempre ha sido una de las prioridades en el desarrollo de la tecnología Bluetooth y continúa siéndolo. La especificación Bluetooth ofrece tres modos de seguridad.
- El coste de los chips Bluetooth es inferior a tres dólares estadounidenses.

La siguiente tabla muestra una comparación de las características técnicas de las diferentes tecnologías inalámbricas del mercado [22]. En ella se aprecian los anteriormente mencionados bajos valores de la tecnología Bluetooth en los campos de precio, consumo de energía y alcance.

Característica	ZigBee	Bluetooth	802.11b	802.11g	802.11a	802.11n	UWB
Rendimiento (Mbps)	0.03	1-3	11	54	54	200	200
Alcance máximo (m)	23	10	60	60	45	45	10
Energía (mW)	30	100	750	1000	1500	2000	400
Ancho de banda (MHz)	0.6	1	22	20	20	40	500
Precio (\$)	2	3	5	9	12	20	7

Tabla 3: Comparativa tecnologías inalámbricas

2.4.4 Arquitectura de Protocolos Bluetooth

Uno de los principales objetivos del planteamiento de la tecnología Bluetooth es conseguir que aplicaciones de diferentes fabricantes mantengan una comunicación fluida. Para ello, según su filosofía, los distintos sistemas que vayan a cooperar deberán estar implementados sobre una misma estructura de protocolos [23].

- **Pila de Protocolos**

En la Figura 3 se muestra la pila de protocolos completa definida en la especificación de Bluetooth.

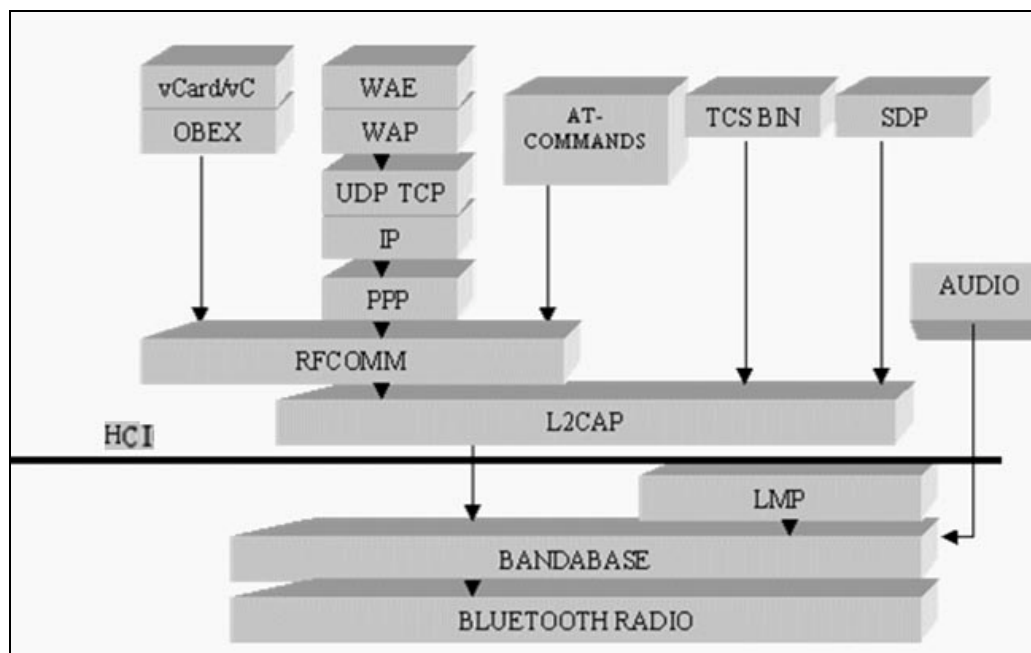


Figura 3: Pila de protocolos Bluetooth

En el diseño de esta arquitectura se ha intentado maximizar el número de aplicaciones que puedan implementarse e interactuar con Bluetooth. Para ello, se han reutilizado los protocolos existentes en capas superiores, siendo específicos de Bluetooth los de nivel más bajo. Normalmente, las aplicaciones no hacen uso de todos los protocolos de la pila. Sin embargo, la especificación es abierta y por tanto permite a los fabricantes desarrollar sus propios protocolos de aplicación para crear un mayor número de aplicaciones que usen las capacidades de Bluetooth.

La pila de la figura está compuesta por numerosos protocolos que pueden ser clasificados según su propósito en los siguientes cuatro grupos:

- **Protocolos centrales de Bluetooth:** Banda Base, LMP, L2CAP, SDP.
- **Protocolos de reemplazo de cable:** RFCOMM.
- **Protocolos de control de telefonía:** TCS BIN, AT-Comands.
- **Protocolos adaptados:** PPP, UDP/TCP/IP, OBEX, WAP, vCard, vCal, WAE).

Los protocolos centrales de Bluetooth son aquellos que han sido completamente desarrollados por el SIG en la mayoría de los dispositivos y realizan las funciones de los niveles físico, enlace y red del modelo de referencia OSI. En cambio, los protocolos de reemplazo de cable, control de telefonía y protocolos adoptados constituyen los orientados a aplicación y su misión es permitir que las aplicaciones funcionen sobre los protocolos centrales de Bluetooth.

La capa de radio es la inferior del estándar y puede ser identificada con el nivel físico. Es la responsable de la transmisión de los datos. Los dispositivos Bluetooth funcionan en la banda de 2,4 GHz, una de las bandas de radio ISM (industrial, científica y médica) que no requieren licencia. Los dispositivos usan un transmisor de salto de frecuencia para contrarrestar las interferencias y la pérdida de intensidad.

La banda base podría englobarse también dentro de la capa física de la torre OSI. Su función principal es controlar los canales y enlaces físicos. Además es la encargada de realizar funciones de sincronización, codificación, decodificación, control de errores y seguridad de bajo nivel. Soporta enlaces de dos tipos SCO y ACL. El primero es un enlace simétrico punto a punto mientras que el segundo es un enlace punto-multipunto.

• Descripción de Protocolos

A continuación, se expone una breve definición de los protocolos de los que se hará referencia a lo largo de este documento.

- **L2CAP o Protocolo de Adaptación y Control de Enlace Lógico:** está implementado sobre el protocolo de Banda Base. Proporciona servicios de datos orientados y no orientados a conexión para los protocolos de

capas superiores. De esta forma, se ocultan a las aplicaciones de alto nivel las particularidades del nivel de enlace y físico. La especificación L2CAP está definida sólo para enlaces ACL.

- **SDP o Protocolo de Descubrimiento de Servicios:** proporciona a las aplicaciones una forma de descubrir los servicios que están disponibles y determinar sus características. Está montado sobre enlaces L2CAP. Una vez que se ha establecido un enlace L2CAP entre dos dispositivos, puede usarse este protocolo para encontrar servicios y conectarse a ellos. Los servicios SDP están compuestos por una serie de atributos. Cada clase de servicio tiene asignado un identificador único, el UUID.
- **RFCOMM o Protocolo de Emulación de Puerto Serie:** se encarga de la emulación de los puertos serie (RS-232) sobre el protocolo L2CAP. Del mismo modo que los puertos serie tienen nueve circuitos, que pueden ser usados para transferir datos y señalización, RFCOMM puede emular sus configuraciones y estados.
- **OBEX o Protocolo de Intercambio de Objetos:** está desarrollado para el intercambio simple y espontáneo de objetos. Usa una estructura cliente/servidor. En su definición incluye un modelo de representación de objetos, un formato estándar para transmitirlos y un protocolo de sesión para el intercambio de peticiones y respuestas entre dispositivos.

Como será posteriormente detallado en el apartado de implementación de la aplicación de este documento, los protocolos utilizados en la aplicación final del proyecto son SDP para buscar los canales de conexión con los dispositivos y RFCOMM para el intercambio de datos.

2.4.5 Perfiles Bluetooth

El estándar Bluetooth se diseñó con el objetivo de ser usado por un amplio conjunto de fabricantes y ser implementado en distintos campos. Para evitar distintas interpretaciones del estándar, el SIG ha definido perfiles Bluetooth [24]. Un perfil define una selección de mensajes y protocolos necesarios para la implementación de una determinada aplicación.

La definición de cada perfil Bluetooth contiene, como mínimo, información acerca de los siguientes conceptos:

- Dependencias con otros perfiles.
- Sugerencias acerca del formato de las interfaces de usuario.
- Partes específicas del protocolo de pila usado por el perfil. Para llevar a cabo esta tarea cada perfil utiliza opciones particulares y parámetros en cada capa de la pila.

La tecnología Bluetooth describe un largo conjunto de perfiles que ilustran diferentes tipos de escenarios donde pueden ser utilizados. En la Figura 4 se muestra los distintos perfiles Bluetooth con su denominación en inglés.

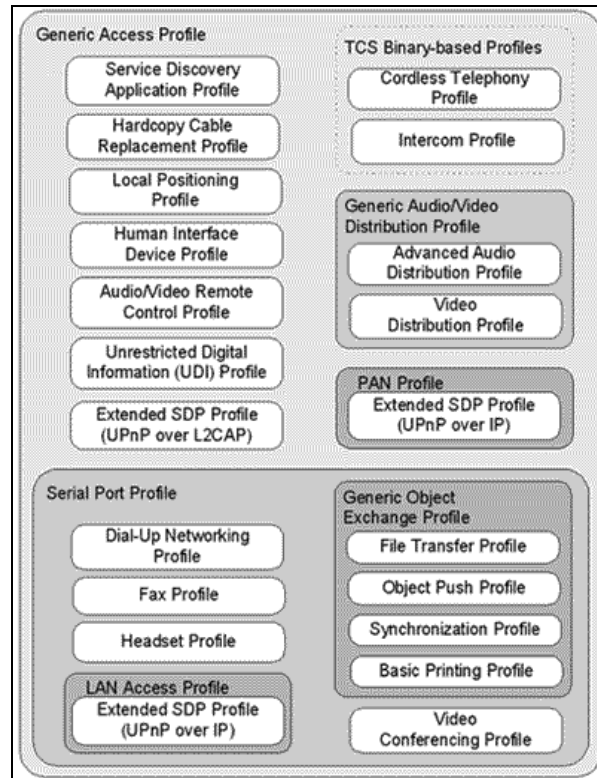


Figura 4: Conjunto de perfiles Bluetooth

Del conjunto de perfiles observados en la figura anterior se hará referencia a lo largo de este documento a los siguientes:

- **HID o Perfil de Dispositivo de Interfaz Humana:** recoge los protocolos, procedimientos y características empleados por las interfaces de usuario *Bluetooth* tales como teclados, dispositivos punteros, consolas o aparatos de control remoto.
- **SPP o Perfil de Puerto Serie:** describe cómo configurar puertos serie y conectar dos dispositivos con tecnología Bluetooth.

2.4.6 Conexiones Bluetooth

Bluetooth proporciona una conexión punto a punto o punto a multipunto. En este último caso, el canal es compartido por varias unidades Bluetooth. Dos o más unidades

que comparten el mismo canal forman una *piconet*. Su formación será imprescindible en la aplicación del proyecto para permitir la conexión con varios sensores simultáneamente.

Una unidad actúa como maestro de la *piconet*, que en el caso concreto de este proyecto sería el dispositivo móvil, mientras que el resto de unidades, los sensores, actúan como esclavos. La norma permite a cualquier equipo asumir cualquiera de los dos papeles, incluso un equipo puede actuar como maestro en un enlace y como esclavo en otro. El maestro es el que se encarga de la sincronización en la *piconet*, estableciendo el patrón de saltos en base a su dirección y reloj.

El número máximo de elementos activos en una *piconet* es de ocho. Como uno asume el rol de maestro, como máximo pueden existir activos siete esclavos. Esto influye directamente en el sistema de la aplicación del proyecto estableciendo este número como el máximo de sensores que se pueden conectar. Además, la *piconet* puede contar con muchos más esclavos en estado de espera, que se mantienen sincronizados con el maestro. Tanto para los esclavos activos como para los aparcados, el acceso al canal es controlado por el maestro.

Múltiples *piconets* con áreas de cobertura solapadas forman una *scatternet*. Cada *piconet* sólo puede tener un maestro. Sin embargo, los esclavos pueden participar en *piconets* diferentes utilizando multiplexación por división en el tiempo. Las *piconets* que forman una *scatternet* no estarán sincronizadas en el tiempo o en frecuencia. Cada una tiene su propio canal con sus correspondientes saltos en frecuencia.

En la siguiente figura se muestran ejemplos de estas estructuras:

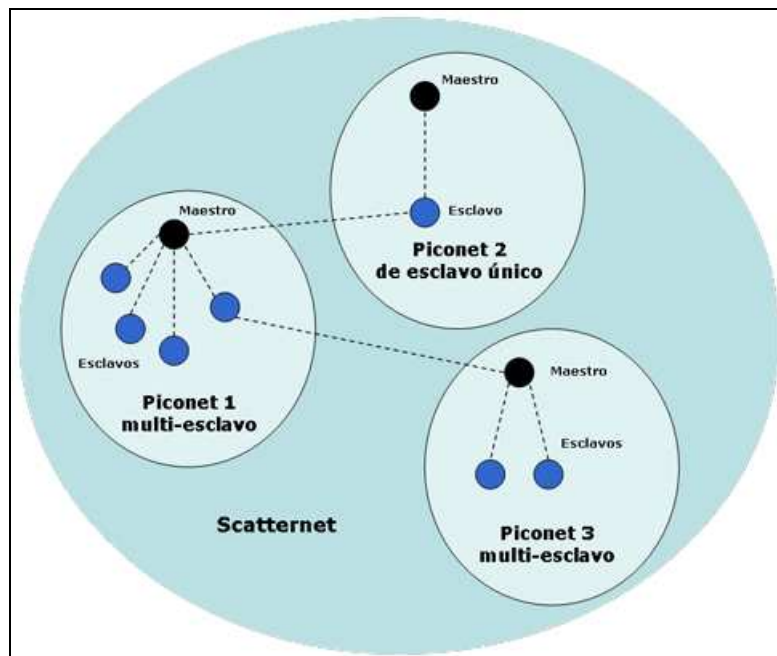


Figura 5: Ejemplos *piconets* y *scatternet*

Dependiendo de la información que el gestor de enlace disponga del dispositivo con el que quiere establecer la conexión se siguen dos procedimientos distintos:

- Si no se conoce nada del aparato remoto debe seguirse primero un procedimiento de búsqueda, denominado *inquiry*, y después otro de asociación, llamado *page*.
- Si se conocen detalles del aparato remoto (dirección física y fase de reloj) tan sólo es necesario llevar a cabo el procedimiento de asociación *page*.

El procedimiento de búsqueda permite a un dispositivo descubrir qué otros equipos se encuentran en su área de cobertura y conseguir la información necesaria para conectarse con ellos. El maestro pregunta si hay dispositivos presentes enviando mensajes IAC si quiere recibir respuesta de cualquier tipo de dispositivo o con un mensaje GIAC si pregunta por algún dispositivo en particular. Un terminal que quiera ser descubierto escuchará el canal y responderá con el valor de su dirección y reloj.

Posteriormente, el procedimiento de asociación permite el establecimiento de la conexión una vez conocido la dirección del dispositivo remoto.

Otro aspecto importante en las comunicaciones inalámbricas es la seguridad. Por ello el estándar de Bluetooth define algoritmos de autenticación y cifrado en el establecimiento de los enlaces.

2.4.7 Dispositivos Bluetooth

En la siguiente figura se pueden ver ejemplos de dispositivos que usan esta tecnología, en concreto, de arriba abajo y de izquierda a derecha se observa unos auriculares de audio, una impresora, un teléfono móvil, una PDA, un mando de videoconsola y un ordenador portátil.



Figura 6: Dispositivos Bluetooth

Cada dispositivo Bluetooth tiene asignada una dirección IEEE MAC de 48 bits conocida como Dirección del Dispositivo Bluetooth. Conjuntamente, cada dispositivo contiene un atributo llamado tipo de dispositivo. Consiste en un valor de 24 bits que el terminal envía, junto con su dirección Bluetooth, como respuesta a una solicitud de búsqueda y que describe el tipo de dispositivo, así como el tipo de sus servicios disponibles. Está formado de la siguiente manera:

- Los últimos 11 bits están reservados para el campo *Major Service Class*. Este describe el servicio soportado por el dispositivo.
- Los siguientes 11 bits están destinados al tipo de dispositivo: 6 últimos bits para *Major Device Class*, el cual describe ampliamente el tipo de dispositivo, y los siguientes 5 bits para *Minor Device Class*, que identifica en detalle el tipo de dispositivo.
- Por último, los primeros 2 bits indican el campo *Format Type*, que condiciona los campos anteriores, pero que por defecto es igual a 0.

Comparando estos valores con los de unas tablas de datos con valores estipulados se obtiene el tipo de dispositivo. Por ejemplo, para identificar el tipo amplio del dispositivo, se debe buscar el valor obtenido en la variable *Major Device Class* en la siguiente tabla:

Valor	Descripción
256	Ordenador
512	Teléfono
768	LAN
1024	Audio/Vídeo
1280	Periférico
1536	Dispositivo de imagen
1892	Complemento que puedes llevar puesto
2148	Juguete

Tabla 4: Valores de Tipo de Dispositivo Bluetooth

2.5 Programación en Dispositivos Móviles

Existen numerosas maneras de realizar y ejecutar programas en dispositivos portátiles. Dependiendo de su sistema operativo, se estudia si cada una de ellas es compatible y finalmente se escoge un entorno y un lenguaje de programación.

Como se comentó anteriormente, en la actualidad se dispone de gran cantidad de dispositivos móviles con altas capacidades de procesamiento y comunicación en el mercado. Además, hay una variedad de lenguajes y entornos de programación que posibilitan el manejo de los mismos.

Para elegir el que mejor se adapta para la resolución del planteamiento del sistema de este proyecto, por un lado se ha hecho un análisis de las posibilidades existentes para la realización de la aplicación, y por otro se han definido unos requisitos necesarios para el programa.

Una aplicación diseñada para un dispositivo móvil puede ser de los siguientes tipos:

- *Stand-alone*: aplicaciones autónomas que se ejecutan en el terminal y no realizan ningún tipo de conexión.
- *Online*: aplicaciones de comunicación entre diferentes dispositivos o desarrollos web para móviles. Pese a que la conexión web no es necesaria, como en la definición del sistema se introduce la necesidad de conectar con varios sensores, se puede encuadrar la aplicación del proyecto en este grupo.

Un problema que se puede encontrar a la hora de realizar una aplicación para un dispositivo móvil es la ausencia de un emulador con las características necesarias. Puesto que un desarrollador programa sobre un ordenador, la aplicación no puede ser ejecutada sobre el mismo aparato, si no que debe existir una migración de la ejecución al dispositivo móvil o a un simulador capaz de ejecutarlo. Si no se dispone de un emulador adecuado, la tarea de transferir la aplicación al dispositivo para su futura ejecución puede resultar tediosa.

Siendo la primera aplicación informática a realizar en un dispositivo móvil, un desarrollador de software puede no sentirse familiar con el nuevo entorno y puede caer en la tentación de adaptar una posible solución para PC a la nueva plataforma. Sin embargo, es necesario tener en cuenta una serie de diferencias que se encuentran tanto en el hardware como en el software de ambos entornos.

2.5.1 Limitaciones Hardware

El primer paso realizado a la hora del diseño de la aplicación ha sido analizar la naturaleza de un dispositivo móvil y las consideraciones a tener en cuenta en base a sus

características y restricciones para diseñar de una manera más efectiva y eficiente la solución de este proyecto.

Estas restricciones hardware pueden ser de varias clases, las cuales son analizadas a continuación:

- **Limitaciones Físicas:**

La primera característica que se aprecia al observar un dispositivo de este tipo es su tamaño. En comparación con un ordenador personal tradicional el dispositivo es bastante más pequeño y por supuesto tanto su pantalla como su capacidad es normalmente mucho más reducida. Por otro lado, el peso es mucho menor lo que lo hace también mucho más manejable.

Ejemplos de dimensiones de dispositivos son los siguientes:

Modelo	Alto (mm)	Ancho (mm)	Grosor (mm)	Peso (g)
Palm T X	120,9	78,2	15,5	149
Asus MyPalA730W	117,5	72,8	16,9	170
Apple iPhone 3G	116	62	12	133
Nokia 5800	111	51,7	15,5	109
T-mobile G1	116,8	54,8	15,7	158

Tabla 5: Dimensiones de dispositivos móviles

- **Limitaciones de Procesamiento:**

La potencia y velocidad de los procesadores de estos dispositivos es más limitada que la de los PC tradicionales. Operaciones que requieran una alta cantidad de recursos de ejecución, como por ejemplo largas búsquedas o lecturas de datos, pueden ralentizar en exceso la ejecución.

Además en aplicaciones con comunicación debe establecerse un equilibrio entre procesamiento y transferencia de datos, es decir, se debe definir si los datos son tratados en el dispositivo, en la red o en el dispositivo con el que se conecta.

- **Limitaciones de Almacenamiento:**

Otra importante restricción es el tamaño de memoria disponible. Mientras que los PC de hoy tienen Gigabytes de almacenamiento en disco y memorias, los dispositivos móviles suelen tener entre 16 y 32 Megabytes de memoria y discos de menor capacidad. El emergente desarrollo de estos dispositivos está acarreado la salida al mercado de productos con almacenamiento mayor. Sin embargo, distan de las cantidades registradas para los PC.

Esta restricción provoca que los programadores tengan que minimizar el conjunto de datos eliminando innecesarias características o apartados de los programas.

Además, el tratamiento de los datos debe ser especialmente minucioso en cuanto a la reserva y liberación de memoria en la propia ejecución de las aplicaciones para evitar limitaciones de memoria y errores en tiempo de ejecución.

Entre otras técnicas de programación, debe limitarse el uso de funciones recursivas y procedimientos que provoque pilas de grandes tamaños. De igual manera, las estructuras de datos deben ser definidas de una manera óptima para reducir el paso de información entre las funciones. La compresión y descompresión de datos puede ser otra técnica a tener en cuenta en este aspecto.

- **Limitaciones de Interfaz:**

Las dimensiones del dispositivo conllevan inherentemente una pantalla de tamaño más reducido que la de un ordenador convencional. Una resolución típica en un ordenador suele variar entre 1024x768 píxeles y 1280x800 píxeles, mientras que en un dispositivo móvil suele tener valores típicos de 240x320 píxeles o menos. Al tener un espacio tan acortado los sistemas operativos no suelen soportar múltiples ventanas o numerosos campos de entrada de datos.

Por esto, la interfaz debe adaptarse adecuadamente a estas dimensiones para que la apariencia de la aplicación sea clara, sencilla, intuitiva y eficiente. Los desarrolladores deben evitar mostrar datos innecesarios en la pantalla y aprovechar correctamente el espacio. Además, debido a su menor resolución, la aplicación tendrá limitaciones en la reproducción de imágenes y vídeos de alta calidad.

Por otro lado y relacionado con las limitaciones de procesamiento, la muestra o el refresco de imágenes y datos deben realizarse únicamente cuando sea estrictamente necesario.

- **Limitaciones de Conectividad:**

Los diferentes modelos de dispositivos ofrecen distintos métodos de comunicación con otros dispositivos.

La sincronización con un ordenador mediante cable, por ejemplo por puerto serie o por USB, es una característica bastante habitual. Sin embargo, las conexiones inalámbricas ofrecidas en la actualidad dependen en gran modo de la modernidad y el precio de los dispositivos debido al desarrollo emergente de las mismas. Entre ellas están los infrarrojos, Bluetooth, Wifi, GSM, GPRS.

Otro aspecto clave a tener en cuenta es la velocidad de transmisión en las comunicaciones inalámbricas. Depende de la tecnología empleada y, por tanto, se debe plantear la cantidad de datos que deben ser transferidos para optimizar el rendimiento. El ancho de banda no depende únicamente del tipo de transmisión disponible, sino también de las condiciones de la red en cada momento. Una aplicación que se adapte correctamente a todas estas posibles condiciones tendrá una usabilidad mayor que la que necesite una transferencia de datos veloz. La compresión y descompresión de los datos

antes de ser transferidos puede ser otra interesante técnica para mejorar el rendimiento de las comunicaciones.

- **Limitaciones de Batería:**

En la mayoría de las ocasiones, los sistemas móviles van a estar desconectados de la corriente eléctrica y por consiguiente el consumo de batería es un concepto básico. Por ello, los desarrolladores deben tratar de emplear la mínima cantidad de energía posible.

Normalmente los sistemas operativos proporcionan métodos de gestión de energía como puede ser la hibernación del sistema cuando el dispositivo no es utilizado. Sin embargo, este proceso de congelación debe ser estudiado e incluso inhabilitado en ocasiones como, por ejemplo, en largas transferencias de datos. Aparte, es recomendable evitar gastos innecesarios de energía, como puede ser un uso ineficiente de memoria, del Bluetooth o una excesiva duración de la ejecución.

Todos estos conceptos deben ser incluidos en el proceso de análisis de la aplicación para optimizar su rendimiento y funcionalidad.

2.5.2 Limitaciones Software

Al contrario de lo que se pueda pensar en principio, la programación para estos dispositivos no consiste en desarrollar aplicaciones de tamaño reducido. Debido a las numerosas limitaciones hardware y sobre todo a las de procesamiento y almacenamiento, es necesario un desarrollo más exhaustivo. Es decir, hay que poner mayor énfasis en las posibles advertencias en la compilación del código, ya que su reproducción y acumulación durante la ejecución pueden provocar problemas. Además, se deben seguir una serie de pautas de programación como evitar código que no va a ser utilizado, funciones auxiliares sin uso real o funciones recursivas que aumenten la complejidad y disminuyan el rendimiento de la aplicación.

Existen pocos libros formales y metodologías para el desarrollo de aplicaciones móviles. Los documentos tradicionales de Ingeniería de Software no tratan este tipo de dispositivos. Sin embargo, siempre es recomendable la utilización de patrones de diseño como por ejemplo el patrón MVC para aplicaciones web.

Debido a la existencia de numerosas plataformas de ejecución para dispositivos móviles, se debe hacer hincapié en la portabilidad de las aplicaciones creadas. En otras palabras, debe hacerse un código reutilizable separando la funcionalidad en módulos para reducir el impacto a la hora de migrar la aplicación a otras plataformas.

Para el acceso y utilización de las distintas capacidades de los dispositivos, existen funciones implementadas y presentadas para su uso en las APIs de los distintos lenguajes de programación existentes. Normalmente, estas APIs son versiones reducidas de las que habitualmente se usan para las computadoras.

2.5.3 Sistemas Operativos

Un sistema operativo es una capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.

Los dispositivos móviles también tienen sus propios sistemas operativos. Con motivo de la existencia de múltiples marcas, modelos y fabricantes de dispositivos móviles, se presenta a su vez la definición de diferentes sistemas operativos que controlan el funcionamiento de los aparatos. Estos son mencionados a continuación:

- **Symbian OS:** es el sistema operativo para móviles más extendido. Fue producto de la alianza de las empresas Nokia, Sony Ericsson, Panasonic, y Siemens. Es un sistema muy robusto, que ahorra siempre que puede batería y espacio en memoria. Con base en Symbian se han desarrollado interfaces de mucho éxito, como la serie 60 de Nokia, que es un estándar abierto.
- **Windows Mobile:** antes conocido como Windows CE, es un sistema operativo empleado tanto en una PDA como en los teléfonos móviles y teléfonos inteligentes de última generación. Fabricantes como Qtek, Samsung, Nec o Palm lo incorporan en sus dispositivos. Una de las ventajas de este sistema es que los programadores pueden desarrollar aplicaciones para móviles utilizando los mismos lenguajes y entornos que emplean con Windows para PC.
- **Blackberry OS:** es la plataforma fabricada por la empresa canadiense Research In Motion para su conocida línea de dispositivos Blackberry. Este dispositivo fue revolucionario al ser el primero en admitir correo electrónico. Aparte, proporciona las capacidades de telefonía móvil, SMS, navegación web y otros servicios de información inalámbricos.
- **Palm OS:** también conocido como Garnet OS, es un sistema operativo fabricado por PalmSource Inc, para PDA. En la actualidad este sistema operativo está en declive ya que la empresa Palm utiliza Windows Mobile en sus nuevos dispositivos y está desarrollando WebOS un programa para aplicaciones abstractas usadas en Internet.
- **iPhone OS:** es el sistema operativo que utiliza el iPhone, el llamativo y exitoso modelo de Apple. Procede del sistema operativo que utilizan los ordenadores personales de la compañía, los Mac. Para realizar aplicaciones existe el iPhone SDK, el cual proporciona un robusto conjunto de herramientas para los desarrolladores, junto con un emulador de software compatible.
- **Android:** es la nueva plataforma libre para teléfonos móviles de Google. Su núcleo está construido sobre Linux, y su código fuente es totalmente nuevo. Permite a los desarrolladores escribir código gestionado en Java y controlar los dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google.

- **Linux:** es el sistema operativo de software libre por antonomasia en el mercado. Fabricantes como Nec, Panasonic o Samsung disponen de numerosos teléfonos con este sistema. El ámbito de las PDA lo lideran las Sharp Zaurus.

En la actualidad, VMWare, empresa que desarrolla software de virtualización, actualmente para PC y Mac ha desarrollado un novedoso proyecto que permite en dispositivos móviles ejecutar varios sistemas operativos para móviles. De esta forma, se puede probar otros sistemas sin necesidad de tener un terminal u otro.

2.5.4 Lenguajes de Programación

De entre los múltiples lenguajes de programación existentes, algunos de ellos han extendido sus características o han creado nuevas versiones para poder aplicarse a los dispositivos móviles. Los más comunes son los siguientes:

- **Java:** es uno de los lenguajes más extendidos para este tipo de dispositivos. Dispone de una especificación de un subconjunto de la plataforma Java orientada a proveer una colección certificada de APIs de desarrollo de software para dispositivos con recursos restringidos, llamada J2ME. Esta orientado a productos de consumo como PDA, teléfonos móviles o electrodomésticos. Su uso está muy extendido en este campo.
- **C/C++:** ideal para el desarrollo de aplicaciones en dispositivos Palm por la API existente para la programación para dispositivos PalmOS.
- **Python:** es un lenguaje de programación interpretado, se desarrolla como un proyecto de código abierto y sus posibilidades son muy amplias. Sin embargo, su uso no está tan extendido como el de otros lenguajes.
- **.NET:** la tecnología de desarrollo de Windows para los sistemas de escritorio se ha extendido a los dispositivos móviles a través de la versión publicada *.NET compact framework*. Permite la programación en los lenguajes C# y Visual Basic. Está orientado a ser ejecutado en dispositivos con sistema operativo Windows Mobile, aunque también se puede ejecutar, por ejemplo, sobre Symbian.

2.6 Programación con Bluetooth

A la hora de realizar aplicaciones que utilicen la tecnología Bluetooth se deben tener en cuenta principalmente dos factores.

El primero consiste en conocer el método de comunicación Bluetooth del dispositivo que va a ejecutar el programa. Este puede ser bien a través del puerto serie o bien a través de *sockets* Bluetooth.

De igual modo, es importante escoger un lenguaje de programación que pueda hacer uso del hardware de Bluetooth del aparato. Actualmente existen numerosos lenguajes con estas características como por ejemplo Java, Python, .NET o C.

2.6.1 Programación mediante Puerto Serie

Un ordenador, bien sea portátil o bien de sobremesa, que en principio carece de conexión Bluetooth, puede conseguir trabajar con esta tecnología mediante el uso de pequeños dispositivos hardware denominados USB dongle. Estos aparatos se conectan a través de un puerto USB de la computadora y permiten la conexión con dispositivos bluetooth utilizando para ello los denominados puertos serie del ordenador también llamados puertos COM.

Para cada dispositivo Bluetooth con el que se quiere conectar, el sistema operativo de la computadora asigna un identificador, el puerto COM, a través del cual se tramitarán las conexiones y el envío y recepción de datos. Estos puertos pueden ser utilizados en las aplicaciones desarrolladas y, por tanto, se puede incluir la tecnología Bluetooth en los programas. Para ilustrar esta técnica de manera práctica, en el Anexo C de este documento se incluye un ejemplo de una aplicación que efectúa una conexión entre un ordenador y un dispositivo Bluetooth a través de un puerto COM.

En cambio, existen dispositivos que tienen conexión cableada por el puerto serie y, al no ser tan complejos como un ordenador, no pueden hacer uso de la opción de conexión indicada anteriormente. Por ejemplo, los dispositivos Palm incluyen la conexión por puerto serie para la sincronización de datos con el ordenador de escritorio. Sin embargo, esta conexión cableada puede ser emulada por Bluetooth. Como se comentó en su definición, el protocolo RFCOMM de Bluetooth se ideó para la emulación de la comunicación por el puerto serie a través del protocolo L2CAP.

Para la utilización del puerto serie en el desarrollo de aplicaciones, los lenguajes de programación normalmente incluyen una serie de librerías que permiten su manejo, tanto si se va a realizar comunicación por cable como si se va a simular por RFCOMM. En el Anexo E, se incluye un ejemplo en el que se crea una aplicación que se ejecuta en el modelo de PDA utilizado para este proyecto. En él se utilizan una serie de funciones pertenecientes a las librerías de programación por el puerto serie obtenidas del API de programación para Palm OS.

Debido a las primeras pruebas realizadas con este tipo de conexión en el dispositivo en el que se iba a ejecutar la aplicación, se descartó el uso del puerto serie para las conexiones Bluetooth. Las librerías ofrecidas para la gestión de la conexión por el puerto serie del modelo Tungsten E2 permiten únicamente la conexión con un dispositivo, independientemente de que se realice por cable o por el protocolo RFCOMM. Esta limitación confronta de manera insalvable con la naturaleza del sistema planteado, el cual debe crear una red Bluetooth, o piconet, y de este modo interaccionar con varios dispositivos simultáneamente.

2.6.2 Programación mediante Sockets Bluetooth

Este método de programación se basa en la comunicación mediante sockets. El término socket designa un concepto abstracto de intercambio de flujo de datos, generalmente de manera fiable y ordenada. La tecnología Bluetooth adapta este concepto para realizar conexiones entre los distintos dispositivos. La mayoría de los terminales Bluetooth, como por ejemplo los teléfonos inteligentes, carecen de puerto serie y, por tanto, este es su método de conexión.

Los tipos de sockets más utilizados en las aplicaciones de esta naturaleza son los de los protocolos SDP, L2CAP y RFCOMM de la pila Bluetooth. Los primeros, los sockets SDP son creados para la búsqueda de servicios de los dispositivos. Su función consiste en solicitar a un dispositivo seleccionado la lista de servicios de conexión que ofrece a otros terminales. En cambio, los otros dos permiten el intercambio de información entre los equipos servidor y cliente. Los sockets L2CAP ofrecen la posibilidad de transferencia de datos en paquetes, mientras que los RFCOMM permite una transferencia de datos serial similar a la realizada por los puertos COM.

Una aplicación tipo que utiliza comunicación mediante sockets Bluetooth normalmente se basa en el siguiente funcionamiento:

1. Se realiza una búsqueda de dispositivos que se encuentren al alcance dentro del radio Bluetooth.
2. De los encontrados se elige un dispositivo para establecer conexión.
3. Mediante el uso de sockets SDP se obtienen los servicios Bluetooth ofrecidos por el dispositivo.
4. Se establece una conexión de datos mediante RFCOMM ó L2CAP entre ambos dispositivos a partir de algún servicio de los encontrados en el paso anterior. Puede necesitarse la creación de un socket de salida y otro de entrada para obtener comunicación bidireccional. El socket SDP ya puede ser cerrado.
5. Se realiza el intercambio de datos necesario en la aplicación.
6. Se cierra el/los socket/s de datos cuando las transferencias concluyan.

El ejemplo incluido en el Anexo D, ilustra de un modo práctico estos pasos necesarios para el establecimiento de una conexión Bluetooth. Para su creación se ha utilizado el API de Java para conexiones Bluetooth estudiado en la siguiente sección de este documento.

Esta técnica permite la conexión simultánea con varios dispositivos repitiendo los pasos anteriormente indicados para cada uno. Además, permite realizar conexiones diferentes que pueden usar cada una un protocolo distinto. Esta característica permite la creación de redes Bluetooth del tipo que se requiere en este proyecto y, por tanto, se trabajará de esta manera en la implementación de la aplicación.

2.6.3 API Java para Bluetooth

El API de Java para Bluetooth es denominado JSR-82 [25] y consiste en un completo conjunto de capacidades que permite una rápida comprensión de las posibilidades de programación con estas conexiones. Está dividido en dos secciones: el paquete `javax.bluetooth` y el paquete `javax.obex`, ambos totalmente independientes entre sí. El primero de ellos define clases e interfaces básicas para el descubrimiento de dispositivos, descubrimiento de servicios, conexión y comunicación. La comunicación a través este paquete es a bajo nivel, es decir, mediante flujos de datos o mediante la transmisión de arrays de bytes. Por el contrario, el paquete `javax.obex` permite manejar el protocolo de alto nivel OBEX (OBject EXchange), utilizado sobre todo para el intercambio de archivos.

Esta programación se basa en la metodología cliente servidor y en el establecimiento de sockets. En una comunicación Bluetooth existe un dispositivo que ofrece un servicio, el servidor, y otros dispositivos acceden a él, los clientes. Las acciones a realizar por el servidor son: crear una conexión servidora, especificar los atributos del servicio, abrir las conexiones clientes y la posterior comunicación. En cambio, las acciones a programar para los clientes son: búsqueda de dispositivos, búsqueda de servicios, establecimiento de la conexión y comunicación.

El paquete `javax.bluetooth` permite usar dos mecanismos de conexión: SPP y L2CAP. Mediante el primero se puede obtener un flujo de datos de entrada (InputStream) y otro de salida (OutputStream). Mediante L2CAP se envían y reciben arrays de bytes.

Para ilustrar cómo funciona este API, y los elementos de que dispone, se ha creado una aplicación de ejemplo que se encuentra detallada en el Anexo D.

Igualmente, el lenguaje Java permite la programación de aplicaciones que utilicen el puerto serie a través de su librería `javax.comm`. Esta técnica ha sido descrita anteriormente en este documento y se plasma mediante el ejemplo práctico del Anexo C.

2.7 Programación para Tungsten E2

A la hora de escoger un método de programación para el dispositivo que, según los requisitos del proyecto, se va a emplear en el sistema, se han considerado todos los aspectos teóricos descritos anteriormente.

- **Java:** la primera opción escogida para realizar aplicaciones en dispositivos móviles debido a que su API está muy desarrollada, a que todo es software libre y a la familiarización con este lenguaje obtenida a lo largo de los estudios universitarios. Mediante la instalación de una máquina virtual de Java para Palm OS, se puede proceder a ejecutar aplicaciones en el dispositivo. Sin embargo, pronto surgió un problema insalvable. La versión de esta máquina virtual compatible con esta PDA no incluye las librerías de Java necesarias para el

manejo de la conexión Bluetooth del dispositivo (jsr-82), impidiendo de este modo el desarrollo de la aplicación. Por tanto, esta plataforma de ejecución no era compatible y fue descartada.

- **C/C++:** es el lenguaje más cómodo para trabajar con dispositivos Palm, puesto que tanto la mayoría de programas que vienen incluidos de fábrica como los disponibles en la red para ser instalados en el dispositivo están realizados en este lenguaje. Además, se puede descargar de la red tanto un entorno de programación como las librerías necesarias para la edición de aplicaciones. Las librerías Bluetooth son perfectamente ejecutables. Será el elegido para codificar la aplicación final.
- **Python:** funciona correctamente en dispositivos Palm y permite el uso de sus capacidades. Sin embargo, no ha sido la opción escogida debido a que se considera el lenguaje C como una alternativa más adecuada por el mayor conocimiento y desarrollo de la misma.

2.7.1 Puerto Serie de Tungsten E2

El puerto serie de un dispositivo Palm puede ser utilizado en la programación de aplicaciones gracias al manejador del puerto serie, denominado Serial Manager, de Palm OS. Es el encargado de controlar la entrada y salida de datos por este puerto así como de la gestión de las señales RS-232, IR, Bluetooth o USB.

Los principales pasos recomendados para la correcta utilización del puerto serie son los siguientes:

- 1) Abrir un puerto serie. Es necesario especificar el identificador del puerto que se desea abrir. De esta operación se obtiene un identificador de la conexión que será utilizado en las operaciones a realizar con el puerto.
- 2) Si es necesario, configurar la conexión. Es posible cambiar cada uno de los parámetros de la configuración por defecto de la conexión. Estos son una cola de recepción de 512 bytes, un tiempo de espera de 5 segundos, control de flujo permitido y bits de 8 datos.
- 3) Realizar el envío o la recepción de datos. Para prevenir la disminución del rendimiento en aplicaciones que utilizan el puerto serie, la recepción de datos se hace de manera asíncrona, mientras que, por el contrario, se realiza un envío síncrono.
- 4) Cerrar el puerto. Una vez que la aplicación ha concluido y que el puerto serie no va a volver a ser utilizado, se debe realizar esta operación para permitir a posteriores aplicaciones su uso y para preservar el estado de la batería del dispositivo.

En el Anexo E de este documento se incluye un ejemplo práctico de una aplicación que utiliza el puerto serie de la PDA para establecer una comunicación Bluetooth.

Como se comentó anteriormente esta técnica de conexión no va a ser utilizada en este proyecto debido a que restringe el número de conexiones únicamente a una.

2.7.2 Sockets Bluetooth de Tungsten E2

El API de Bluetooth para la programación de aplicaciones para PalmOS [30] proporciona a los desarrolladores el camino para realizar programas capaces de hacer uso de las capacidades Bluetooth del dispositivo. Las posibles operaciones que permiten sus librerías son las siguientes:

- Establecer conexiones Bluetooth
- Acceder a una conexión de Internet a través de puntos de acceso LAN o teléfonos móviles.
- Intercambiar objetos a través de Bluetooth.
- Realizar sincronizaciones sin cable con equipos.
- Comunicarse con otros dispositivos en programas multiusuario como juegos o aplicaciones de colaboración.
- Enviar mensajes SMS y gestionar la libreta de teléfonos móviles.

Los sockets Bluetooth implementados en estos dispositivos utilizan conexiones ACL. Las librerías soportan tres tipos de sockets: SDP para la búsqueda de servicios de otros dispositivos y L2CAP y RFCOMM para establecer canales de comunicación y, sobre ellos, realizar el envío y la recepción de datos.

Para ilustrar este método de conexión mediante un ejemplo práctico se ha incluido en el Anexo F de este documento una aplicación que realiza una lectura de datos de un dispositivo externo a través de sockets Bluetooth.

Las conexiones Bluetooth que se van a establecer en el sistema final de este proyecto van a hacer uso de esta técnica de programación. En concreto, se va a crear una piconet de hasta ocho dispositivos y para las comunicaciones se establecerán sockets RFCOMM simultáneos que van a simular la conexión por el puerto serie entre maestro y esclavos.

3. Gestión del Proyecto

En este apartado se realiza una introducción a la metodología llevada a cabo en la realización del proyecto.

En primer lugar, se describen las diversas tareas en las que se ha dividido el trabajo a llevar a cabo. Se ha detallado tanto el objetivo de cada una de ellas como la relación que guardan entre sí.

Posteriormente, se mostrará la duración que ha tenido cada una de estas tareas definidas y su consiguiente gasto económico. Además, se ha ilustrado la completa realización del proyecto a lo largo del tiempo dedicado con la ayuda de un diagrama de Gannt.

Finalmente, se incluye el presupuesto del proyecto en el que se incluye el coste final.

3.1 Descomposición en Tareas

La entera realización del proyecto se ha dividido en una serie de fases establecidas a partir del planteamiento del problema a resolver. Cada una de estas tareas está compuesta por un conjunto de estudios y actividades a llevar a cabo. El desarrollo de las tareas ha constituido la evolución progresiva del proyecto y éstas se han realizado de acuerdo con un orden estipulado con la intención de obtener una correcta planificación del proyecto. A continuación, se detallan las fases en las que se ha repartido la realización del proyecto.

3.1.1 Actividad A: Análisis del Problema

El primer procedimiento que se ha realizado ha sido el del análisis del planteamiento del problema con el fin de adquirir una idea más detallada de qué es lo que se quiere hacer. Basándose en este objetivo, se han realizado varias reuniones con el profesor en las que se han detallado características que debe tener la aplicación y

requisitos de funcionalidad. A partir de este estudio, se han definido los diferentes dispositivos que se van a utilizar, así como las necesidades tecnológicas para la comunicación entre ellos.

3.1.2 Actividad B: Documentación y Análisis del Estado del Arte

Esta fase consiste en la investigación acerca del estado actual de las tecnologías y dispositivos que van a ser utilizados en la realización del proyecto.

- **Tarea B1: Investigación Teórica de las Tecnologías**

A partir del primer estudio del sistema deseado, se ha realizado un análisis del estado actual de los dispositivos y las tecnologías necesarias para la elaboración del proyecto. Se ha compuesto por una investigación puramente teórica de los términos y conceptos extraídos en la fase anterior así como por una búsqueda de ejemplos similares existentes realizados por terceras personas. Este proceso ha sido resumido en el apartado de estudio del arte de este documento, respetando el orden que se ha seguido en el proceso.

- **Tarea B2: Estudio de Viabilidad**

De la mano de la anterior etapa, se ha realizado un estudio de la viabilidad del sistema. Es decir, se ha analizado tanto si lo que se plantea en el enunciado del problema es factible y puede ser implementado con los recursos que se disponen en la actualidad, como los requisitos físicos y temporales necesarios. Para realizar este proceso se ha utilizado la información obtenida en la investigación previa.

- **Tarea B3: Realización de Aplicaciones de Prueba**

A partir del conocimiento teórico adquirido en las anteriores tareas, se han realizado sencillas aplicaciones informáticas para comprobar el correcto funcionamiento de los dispositivos y contemplar las diferentes posibilidades de programación del sistema final. Estas aplicaciones desarrolladas se han documentado en los anexos de este documento.

Esta fase incluye además la elección del método de desarrollo que se ha llevado a cabo en la implementación de la aplicación y la familiarización con los distintos dispositivos y herramientas de desarrollo. Con este objetivo se ha probado el funcionamiento de los sensores mediante el uso de las aplicaciones ya existentes. A continuación, se han instalado diferentes programas en el dispositivo móvil para la comprobación de sus funcionalidades. Finalmente, se han establecido conexiones

Bluetooth entre los diferentes dispositivos para analizar el correcto funcionamiento de la comunicación.

3.1.3 Actividad C: Diseño de la Aplicación

En esta fase se ha diseñado la aplicación final que llevará a cabo la ejecución del sistema. Se ha elaborado teniendo en cuenta los requisitos del planteamiento del problema. Al mismo tiempo, se han aplicado los conocimientos tecnológicos e informáticos adquiridos a lo largo de los estudios universitarios para realizar un diseño más adecuado. Se ha hecho hincapié no solo en la resolución del problema sino también en la calidad del desarrollo, la robustez de la aplicación y la posibilidad de reutilización de módulos del sistema.

Esta fase ha sido ilustrada en el apartado de diseño de la aplicación de este documento, en el que se han incluido diagramas para su más rápida y mejor comprensión.

- **Tarea C1: Diseño de la Interfaz de Usuario**

Por un lado, se ha diseñado el aspecto de la interfaz de usuario así como los elementos que debe contener. Se ha analizado la información que debe ser mostrada por pantalla y la interacción a llevar a cabo con el usuario. Además, se han considerado las posibilidades y limitaciones del dispositivo en este aspecto.

- **Tarea C2: Diseño del Tratamiento de Datos**

En primer lugar, esta tarea consiste en identificar el formato de los datos que los van a enviar los diferentes terminales con los que se va a conectar el dispositivo central del proyecto. Seguidamente, se ha diseñado la base de datos del dispositivo que almacenará y manejará los datos capturados. Finalmente, se estudia el método de transferencia de los datos almacenados a otro dispositivo donde se alojarán de manera permanente.

- **Tarea C3: Diseño de la Lógica de la Aplicación**

Mediante un modelo de diseño por capas, se ha planteado la división de la lógica del sistema en módulos con el fin de separar la funcionalidad de la aplicación en subtarefas que sean implementadas de una manera más sencilla y progresiva.

Se han desarrollado diagramas UML para obtener un diseño acorde con los requisitos de modelado que ofrece este conocido lenguaje.

3.1.4 Actividad D: Implementación de la Aplicación

Posteriormente a la realización final del diseño, se ha procedido a la implementación de la aplicación. Esta fase consiste en la creación del programa informático final que será ejecutado para la resolución de la propuesta del proyecto. Por tanto, engloba tanto la realización de la interfaz de usuario como la codificación de la aplicación mediante un lenguaje de programación.

Se ha documentado esta fase en el apartado de implementación de la aplicación de este documento.

- **Tarea D1: Creación de la Interfaz de Usuario de la Aplicación**

Mediante la herramienta de desarrollo de interfaces de usuario utilizada, se han generado los distintos formularios que serán mostrados por pantalla y los diferentes elementos que componen cada uno de ellos. Esta tarea ha sido desarrollada conjuntamente con la implementación del código fuente de la aplicación para obtener diferentes prototipos del sistema final que han ido aumentando en complejidad.

- **Tarea D2: Implementación del Código Fuente de la Aplicación**

La implementación se ha llevado a cabo de una manera progresiva, es decir, mediante la ejecución y comprobación de los diferentes módulos diseñados. Por tanto, se han creado prototipos de la aplicación que han sido evaluados y a los que se iba añadiendo la complejidad necesaria para acabar completando la funcionalidad requerida en el sistema.

3.1.5 Actividad E: Evaluación de la Aplicación

La fase de evaluación consiste en realizar una serie de pruebas de diferente nivel de complejidad para comprobar el correcto funcionamiento de la aplicación programada y obtener conclusiones acerca del resultado final del proyecto.

Se ha incluido información acerca de esta fase en el apartado de evaluación de la aplicación de este documento.

- **Tarea E1: Definición de las Pruebas**

Como se ha comentado anteriormente, dentro del conjunto de las pruebas realizadas, existen diferentes niveles de complejidad. Algunas de ellas fueron definidas en el planteamiento del proyecto debido a que consisten en comprobar si la funcionalidad del sistema es completa, es decir, si realiza las operaciones que se requerían en un principio de manera correcta.

Por otro lado, se definen los denominados tests de calidad en los que se comprueba y analiza el rendimiento de la aplicación.

- **Tarea E2: Realización de las Pruebas**

Mediante el uso de los distintos dispositivos y a partir del planteamiento anterior de las diferentes situaciones de evaluación del sistema, se ha llevado a cabo el conjunto de pruebas. De esta manera se ha recogido una serie de resultados numéricos que han sido almacenados para su análisis.

- **Tarea E3: Análisis de los Resultados**

Por último, se han extraído unas conclusiones acerca de la calificación de las distintas evaluaciones realizadas y del resultado global de las ejecuciones del programa.

3.1.6 Actividad F: Redacción de la Memoria

El último proceso en la realización del proyecto ha sido la redacción de este documento. Esta fase ha sido desarrollada a lo largo de todas las etapas anteriores, debido a que cada una de ellas ha sido debidamente documentada.

3.2 Duración

En la siguiente tabla se muestra la duración de las tareas definidas en el apartado previo.

Tarea	Duración (días)	Dedicación (horas/día)	Total Horas
A. Análisis del problema	2	4	8
Total Actividad A			8
B. Documentación y Análisis del Estado del Arte			
B1. Investigación Teórica de las Tecnologías	27	4	108
B2. Estudio de Viabilidad	25	2	50
B3. Realización de Aplicaciones de Prueba	20	2	40
Total Actividad B			198
C. Diseño de la Aplicación			
C1. Diseño de la Interfaz de Usuario	6	5	30
C2. Diseño del Tratamiento de Datos	1	6	6
C3. Diseño de la Lógica de la Aplicación	18	6	108
Total Actividad C			144
D. Implementación de la Aplicación			
D1. Creación de Interfaz de Usuario de la Aplicación	6	5	30
D2. Implementación del Código Fuente de la Aplicación	17	6	102
Total Actividad D			132
E. Evaluación de la Aplicación			
E1. Definición de las Pruebas	5	2,2	11
E2. Realización de las Pruebas	79	1,2	94
E3. Análisis de los Resultados	69	0,3	21
Total Actividad E			126
F. Redacción de la memoria	164	0,5	82
Total Actividad F			82
Total Proyecto			690

Tabla 6: Duración de las tareas del proyecto

3.3 Diagrama de Gannt

En el diagrama de la Figura 7 se enumeran las distintas fases del proyecto anteriormente expuestas. Además, en la imagen posterior se muestra de una manera

gráfica la duración de las mismas y la relación existente entre ellas a la hora de haber sido desarrolladas.

Nombre	Fecha de inicio	Fecha de fin	Duración
Análisis del problema	3/11/08	5/11/08	2
Documentación y análisis del estado del arte	5/11/08	20/12/08	33
Investigación teórica de las tecnologías	5/11/08	13/12/08	28
Estudio de viabilidad	13/11/08	18/12/08	25
Realización de aplicaciones de prueba	24/11/08	20/12/08	20
Diseño de la aplicación	19/12/08	14/01/09	18
Diseño de la interfaz de usuario	6/01/09	14/01/09	6
Diseño del tratamiento de datos	13/01/09	14/01/09	1
Diseño de la lógica de la aplicación	19/12/08	14/01/09	18
Implementación de la aplicación	20/01/09	14/02/09	19
Creación de la interfaz de usuario de la aplicación	6/02/09	14/02/09	6
Implementación del código fuente de la aplicación	20/01/09	12/02/09	17
Evaluación de la aplicación	2/03/09	26/06/09	84
Definición de pruebas	2/03/09	7/03/09	5
Realización de pruebas	9/03/09	26/06/09	79
Análisis de los resultados	23/03/09	26/06/09	69
Redacción de la memoria	10/11/08	26/06/09	164

Figura 7: Listado de tareas del proyecto

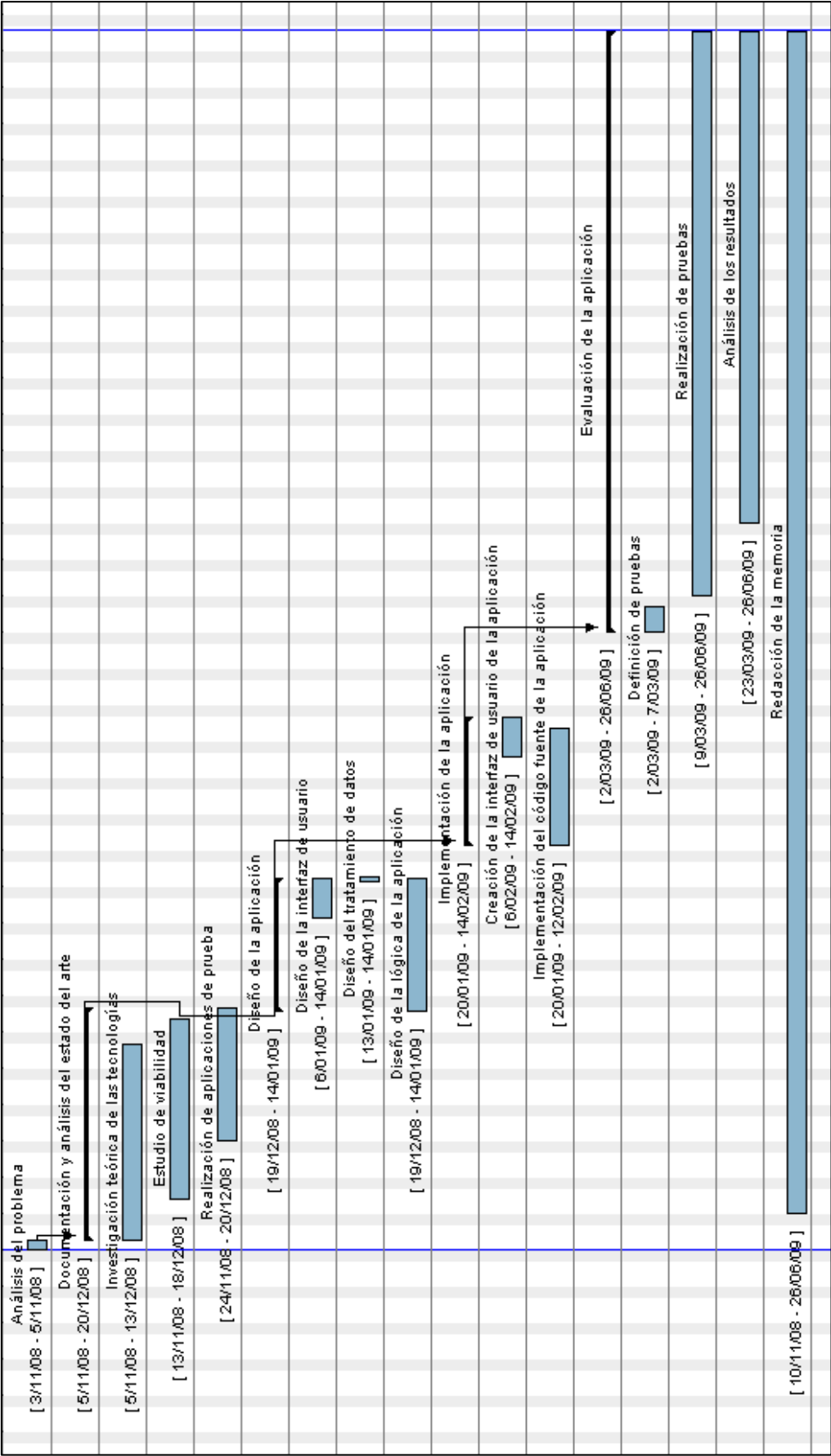


Figura 8: Diagrama de Gantt

3.4 Presupuesto

En esta sección se incluyen los costes totales del proyecto descompuestos en costes de personal y de equipamiento. Para evitar confusiones se debe comentar que todas las cantidades están expresadas en euros.

3.4.1 Costes de Personal

A continuación se encuentra desglosado el coste personal del proyecto según las diferentes actividades realizadas y previamente descritas. El precio por hora del trabajo de un ingeniero superior de informática se ha estimado de 30 euros.

Tarea	Duración (horas)	Coste (€)
A. Análisis del problema	8,0	240,00
B. Documentación y Análisis del Estado del Arte	198,0	5.940,00
C. Diseño de la Aplicación	144,0	4.320,00
D. Implementación de la Aplicación	132,0	3.942,00
E. Evaluación de la Aplicación	126,0	3.780,00
F. Redacción de la memoria	82,0	2.460,00
Total Actividades	690,0	20.682,00

Tabla 7: Costes de personal

3.4.2 Costes de Equipamiento

En este apartado se incluye la descripción del gasto realizado en el material empleado en el proyecto.

Equipo	Coste (€)
Ordenador portátil HP Pavilion DV7-1145ES	699,00
USB Dongle Belkin	15,00
PDA Tungsten E2	199,00
Sensores de aceleración WiTilt v2.5 (4 unidades)	600,00
Sensor RFID	149,00
Adaptador Bluetooth RS232	81,95
Teléfono móvil Nokia 6120	182,00
Mando de la videoconsola Wii	30,00
Antena GPS Bluetooth	30,00
Total equipamiento	1.985,95

Tabla 8: Costes de equipamiento

3.4.3 Coste Total

Finalmente, se calcula el coste total del proyecto.

- **Costes de personal 20.682,00 euros**
- **Costes de equipamiento 1.985,95 euros**
- **Subtotal 22.667,95 euros**
- **Impuestos (16% de IVA) 3.626,87 euros**
- **Total 26.294,82 euros**

El coste total del proyecto asciende a veintiséis mil doscientos noventa y cuatro euros y ochenta y dos céntimos de euro.

4. Memoria del Trabajo Realizado

En este apartado se describe el desarrollo de la aplicación. Partiendo de una fase de análisis en la cual se introducen una serie de requisitos que debe cumplir el programa final, se estudian las distintas fases por las que debe transcurrir la lógica de la aplicación.

A continuación, se redacta el diseño de la aplicación. Se introducen los distintos elementos que va a tener la aplicación y se divide la lógica de la misma en módulos. Además, se incluyen diagramas comentados para una completa comprensión.

Finalmente, se describe la implementación del diseño en el programa final. Se detalla los diferentes métodos escogidos para su programación y se muestran imágenes con la apariencia final de la aplicación.

4.1 Consideraciones Previas al Diseño de la Aplicación

Del planteamiento del problema que la aplicación debe solucionar se ha extraído una serie de información que se resume en este apartado. Se introducen consideraciones acerca de los recursos hardware y software que debe tener el sistema y en especial el dispositivo móvil central para el correcto funcionamiento de la aplicación. De los comentarios recogidos a continuación se obtienen unas condiciones que debe tener el sistema y que deben ser comprobadas en la fase de evaluación de la aplicación.

- Para la conectividad del sistema se necesita emplear el Bluetooth del dispositivo. Por lo tanto, en la programación se tendrá que incluir el manejo de las librerías Bluetooth del lenguaje. Además, un requisito clave de la aplicación es la posibilidad para establecer más de una conexión Bluetooth simultánea, y por ello, deberá crearse una *piconet*.
- Los datos obtenidos de las lecturas de los sensores deberán ser almacenados de alguna manera en el dispositivo para permitir su posterior tratamiento. Esta escritura debe ser permanente, es decir, no deben borrarse los datos aunque se finalice la ejecución del programa o incluso se apague el dispositivo.

- Los recursos de funcionamiento necesarios no son complejos puesto que los procedimientos que se realizarán serán bastante sencillos. Es decir, no se van a realizar operaciones complicadas como pueden ser recursividad u operaciones aritméticas complejas.
- La interfaz no requiere la entrada de un gran volumen de datos, pero debe estar capacitada para mostrar una gran cantidad de información. Se compondrá de formularios en los que se representarán las diferentes etapas de la aplicación para su cómodo seguimiento.
- Es necesario un control de la memoria del dispositivo necesaria para las ejecuciones. Los datos recibidos de los sensores ocupan unos pocos bytes por lectura. Sin embargo, la cantidad de datos procesados aumentará proporcionalmente al número de sensores conectados y al número de repeticiones.
- El consumo de la batería dependerá de la duración de cada ejecución. Debe ser imprescindible evitar que el dispositivo se apague durante la recepción de datos de los sensores para no provocar errores en la aplicación.

4.2 Fases de la Aplicación

Para realizar la división en fases del sistema, se ha realizado el diagrama de casos de uso de la siguiente figura.

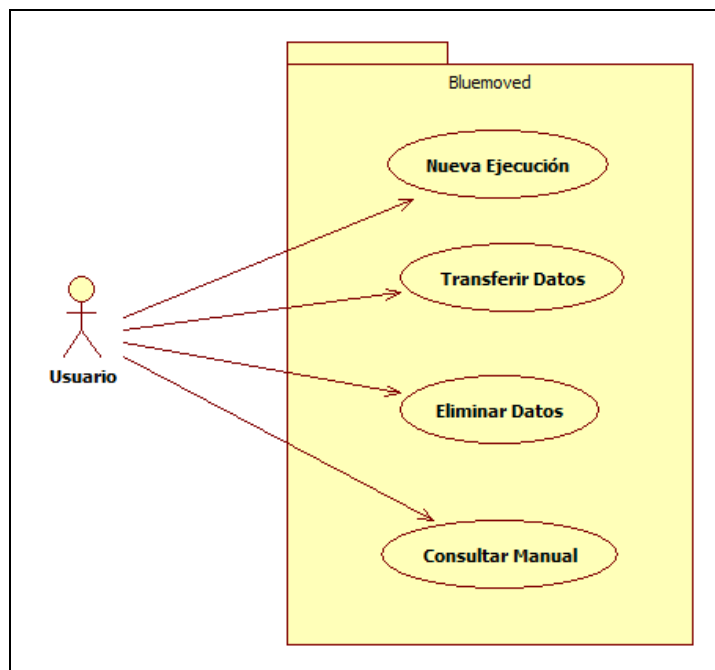


Figura 9: Diagrama de casos de uso

En él se observa la existencia del usuario y las actividades que puede realizar en el programa. Estas actividades son independientes y, por tanto, se creará un menú principal de la aplicación desde el cual se pueda acceder a cualquiera de ellas. Los casos de uso del diagrama son brevemente descritos a continuación:

- **Nueva Ejecución:** es el más completo y supone la realización de principio a fin de una lectura de datos de los sensores. Por su complejidad, ha sido dividida en las siguientes subtarear: configuración, conexión, captura y resultados.
- **Transferir Datos:** consiste en la transferencia de los datos almacenados en las lecturas a otro dispositivo externo para su futuro tratamiento.
- **Eliminar Datos:** realiza directamente un borrado de los datos almacenados en lecturas previas.
- **Consultar Manual:** se le muestra al usuario, a través de la pantalla del dispositivo, la información necesaria para llevar a cabo una correcta ejecución de la aplicación.

Por consiguiente, el diseño se realiza por fases de funcionamiento que son implementadas por separado y en su implementación van formando prototipos del sistema. Estas fases están descritas a continuación.

4.2.1 Menú

Se mostrará un menú al usuario con las siguientes opciones:

- a) **Iniciar ejecución:** inicia la ejecución de un proceso de lectura de datos de los sensores. Se procederá a la configuración (apartado 4.2.2).
- b) **Transferir datos:** realiza una transferencia de los datos almacenados en la BBDD de la aplicación a otro dispositivo (apartado 4.2.6).
- c) **Borrar BBDD:** borra automáticamente todas las entradas de la BBDD sin realizar ninguna transferencia previa.
- d) **Manual de usuario:** muestra información relacionada con el funcionamiento de la aplicación y consejos frente a la aparición de una serie de problemas frecuentes.
- e) **Salir:** finaliza la ejecución del programa volviendo al menú del Sistema Operativo.

4.2.2 Configuración

Se solicitará al usuario que introduzca valores para los siguientes parámetros de configuración de la aplicación:

- **Número de dispositivos:** se permitirá un valor mínimo de 1 y máximo de 7 debido a las restricciones de redes de Bluetooth.
- **Frecuencia de captura de datos:** se permitirá escoger de una lista un valor entre las siguientes opciones: 100 y 50 Herzios.

4.2.3 Conexión con los Dispositivos

Esta fase consiste en la parte crítica del programa debido a que es la que más problemas puede provocar. Se procederá a establecer las conexiones con el número de dispositivos que el usuario ha introducido en la fase de configuración.

Para comenzar, se mostrará al usuario una lista de dispositivos que la PDA es capaz de encontrar en su radio Bluetooth. Se solicitará al usuario la elección de los dispositivos. Posteriormente, se realizarán las conexiones de manera transparente al usuario y se mostrará información relacionada con el resultado de cada conexión.

En este apartado pueden surgir problemas relacionados con el número de dispositivos encontrados, la creación de los enlaces o el establecimiento de las conexiones. Si esta operación finaliza con éxito, se permite al usuario comenzar con la lectura de datos.

4.2.4 Captura de Datos

En esta fase se produce la lectura de datos de los dispositivos con los que se han establecido las conexiones.

El usuario tendrá la opción de visualizar los datos en tiempo real. Sin embargo, esta opción puede ralentizar el proceso y retardar la frecuencia de recogida de datos. Se posibilitará al usuario finalizar este proceso en cualquier momento. A su término, se mostrará una serie de datos resumen del proceso. Además, los datos se guardarán en la base de datos de la aplicación cuya especificación está detallada posteriormente.

4.2.5 Muestra de Resultados

En este apartado se mostrará al usuario un resumen de la ejecución del proceso de lectura de datos. Se indicará si la operación se ha desarrollado y ha concluido con éxito o si ha ocurrido algún error.

Además se incluirá el número de dispositivos conectados, el número de lecturas realizadas y la duración de la operación.

4.2.6 Transferencia de Datos

Esta fase consiste en la transmisión de los datos desde el dispositivo portátil a otro dispositivo electrónico para el posterior vaciado de la base de datos.

Al igual que en la fase de conexión, se ofrecerá una lista de los dispositivos que la PDA encuentra en su radio Bluetooth. El usuario elegirá el dispositivo al que desee transferir los datos y se mostrará bien un mensaje de éxito al finalizar la operación o bien los errores producidos durante la misma. Puede realizarse la transferencia de ejecuciones pasadas aunque el programa haya sido cerrado o incluso que el dispositivo haya sido apagado.

4.2.7 Borrado de Datos

En esta fase se realiza un vaciado de la base de datos del dispositivo local, es decir, se borran todos los datos obtenidos de ejecuciones.

4.2.8 Manual de Usuario

Esta fase consiste en la realización de un manual que el usuario pueda consultar, directamente en el dispositivo, y que le sirva de ayuda para realizar ejecuciones correctas.

4.3 Diseño de la Aplicación

En este apartado se realiza una descripción del diseño de la aplicación elaborado previamente a su implementación.

Para ello, se ha tenido en cuenta los distintos aspectos de la programación de un dispositivo portátil como es la Palm Tungsten E2, analizados en capítulos anteriores, puesto que pueden condicionar el desarrollo del programa.

4.3.1 Formato de la Aplicación

La aplicación será identificada mediante un nombre y un icono característico para ser reconocida de una manera rápida y sencilla. El nombre de la misma será “Bluemoved2” y el icono será la siguiente figura:

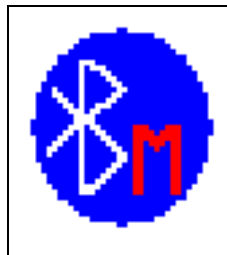


Figura 10: Icono de la aplicación

Por otro lado la aplicación será alojada en una carpeta llamada “PFC” para tener un acceso directo desde el menú de la PDA.

El texto estará escrito en lengua española y las imágenes estarán definidas para pantallas de color.

4.3.2 Modelo de Diseño en Tres Capas

Además de dividir la funcionalidad de la aplicación en fases, se va a separar la lógica del funcionamiento del programa en varios módulos o sistemas que interconecten entre sí para intercambiarse la información.

Para optimizar en mayor modo la división del trabajo de desarrollo se ha escogido un modelo de diseño de tres capas ilustrado en la Figura 11.

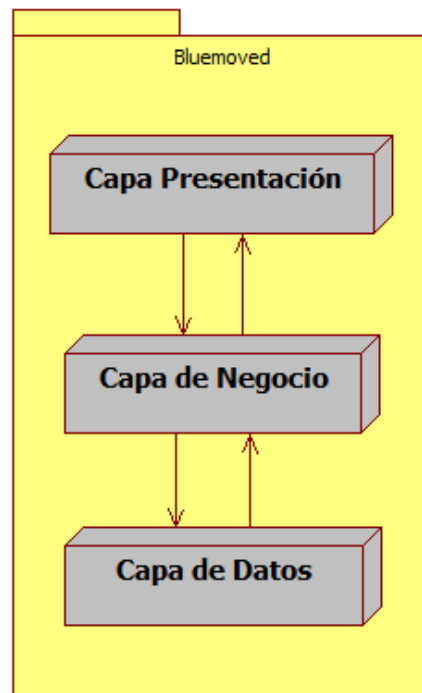


Figura 11: Modelo de programación tres capas

La capa superior, la denominada Capa de Presentación, es la que gestiona todo lo relacionado con la interfaz de usuario, tanto lo que se ha de mostrar en el dispositivo como lo que se recibe por interacción con el usuario. La segunda capa, denominada Capa de Negocio, es la encargada de la gestión de la lógica de la aplicación y, por tanto, del flujo de datos y procesos que realiza la aplicación. Por último, la capa más baja, denominada capa de datos, es la encargada de manejar la base de datos de la aplicación tanto para peticiones de lectura como de escritura. Entre las tres existe comunicación que se realiza a través de interfaces como se muestra en el diagrama de componentes de la siguiente figura.

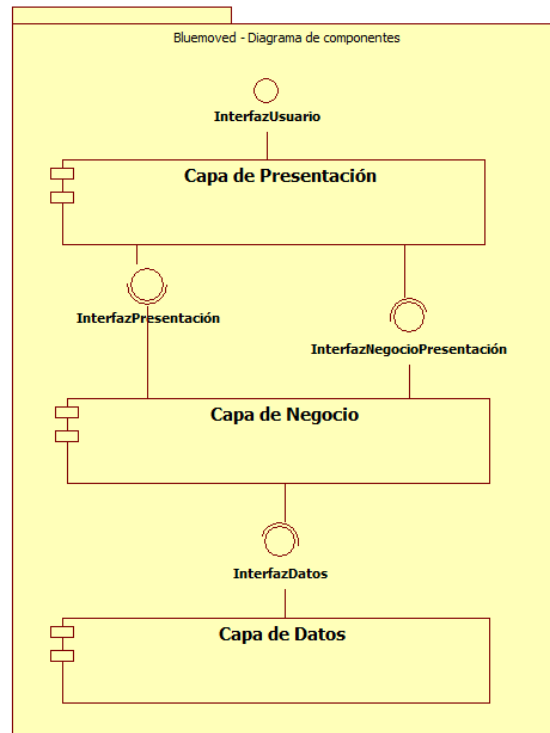


Figura 12: Diagrama de comunicación entre capas

Como se puede apreciar existen tres interfaces de comunicación:

- **InterfazPresentacion** que es ofrecida por la Capa de Presentación y accedida desde la Capa de Negocio.
- **InterfazNegocioPresentación** que posibilita la comunicación en el sentido inverso a la anterior.
- **InterfazDatos** que permite la comunicación de la Capa de Negocio con la capa de Datos y, por tanto, con la base de datos de la aplicación.

4.3.3 Capa de Presentación

Esta capa realiza dos funciones generales basadas en la gestión de la interfaz de usuario. Por un lado recibe la información proporcionada por el usuario a través de los elementos de entrada de datos. Por otro lado, muestra por pantalla el estado y los resultados de la aplicación.

Se compone de la clase Presentacion. Además, ofrece la interfaz de comunicación InterfazPresentacion que es implementada por ella. La composición de ambas se puede observar en el diagrama de clases de la siguiente figura.

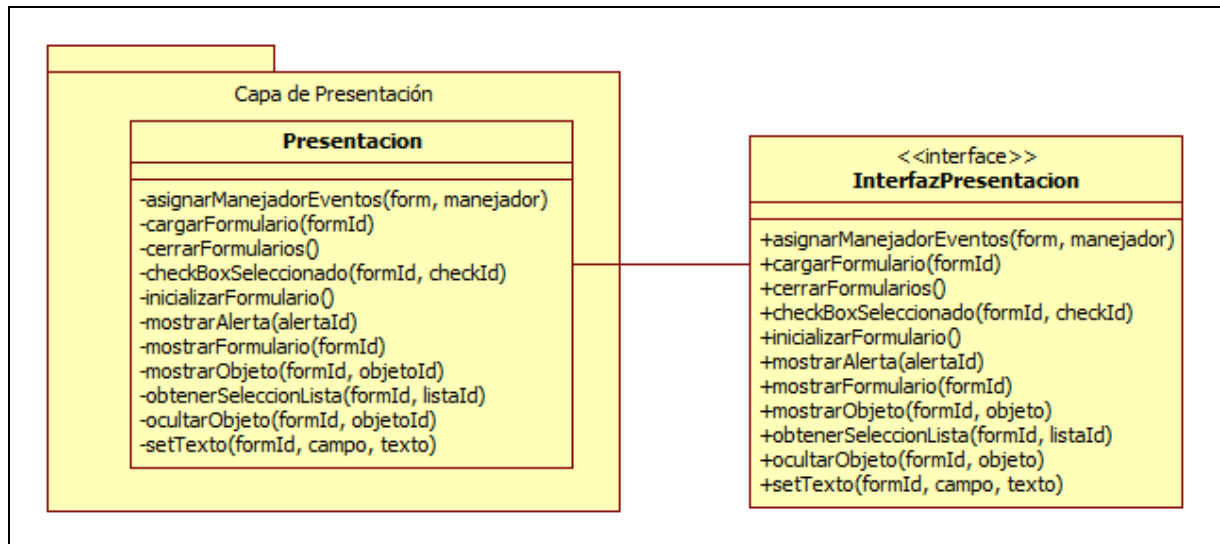


Figura 13: Diagrama de clases Capa Presentación

Ambas entidades poseen las mismas funciones debido a que la clase *Presentacion* implementa las funciones ofrecidas por la interfaz *InterfazPresentacion* a la Capa de Negocio. Como se puede observar en el diagrama las funciones de la clase *Presentacion* son de acceso privado mientras que las de su interfaz son públicas.

Las operaciones son las básicas de tratamiento de la interfaz de un dispositivo móvil y son descritas a continuación:

- **asignarManejadorEventos(form, manejador):** establece una función manejador, perteneciente a la capa de negocio, como la encargada de manejar los eventos producidos en el formulario form.
- **cargarFormulario(formId):** dispone el formulario cuyo identificador es formId para ser utilizado.
- **cerrarFormularios():** cierra todos los formularios abiertos debido a que la aplicación va a concluir.
- **checkBoxSeleccionado(formId, checkId):** obtiene si el *checkbox* cuyo identificador es checkId está seleccionado por el usuario o no y lo devuelve como valor numérico (1 si está seleccionado y 0 en caso contrario).
- **inicializarFormulario():** muestra por pantalla el formulario que se encuentra disponible para ser utilizado.
- **mostrarAlerta(alertaId):** muestra por pantalla la alerta cuya identificador es alertaId.
- **mostrarFormulario(formId):** muestra por pantalla el formulario cuyo identificador se recibe por parámetros.
- **mostrarObjeto(formId, objetoId):** muestra por pantalla, en el formulario cuyo identificador es formId, el objeto identificado por objetoId.

- **obtenerSeleccionLista(formId, listaId):** devuelve la posición que ocupa el elemento seleccionado de la lista identificada por listaId.
- **ocultarObjeto(formId, objetoId):** esconde el objeto cuyo identificador es objetoId y, por tanto, deja de ser mostrado en el formulario formId.
- **setTexto(formId, campo, texto):** inserta el texto obtenido por parámetros en el campo correspondiente en el formulario cuyo identificador es formId. Además, refresca la pantalla para mostrar el nuevo valor.

4.3.4 Capa de Negocio

La capa de negocio es la encargada de controlar toda la lógica de la ejecución de la aplicación. Es, por tanto, la capa más compleja y para simplificar su desarrollo se divide el trabajo en subtarear en base a la funcionalidad que debe tener el sistema. Estas subtarear se han agrupado en las clases que conforman el diagrama de clases de la siguiente figura.

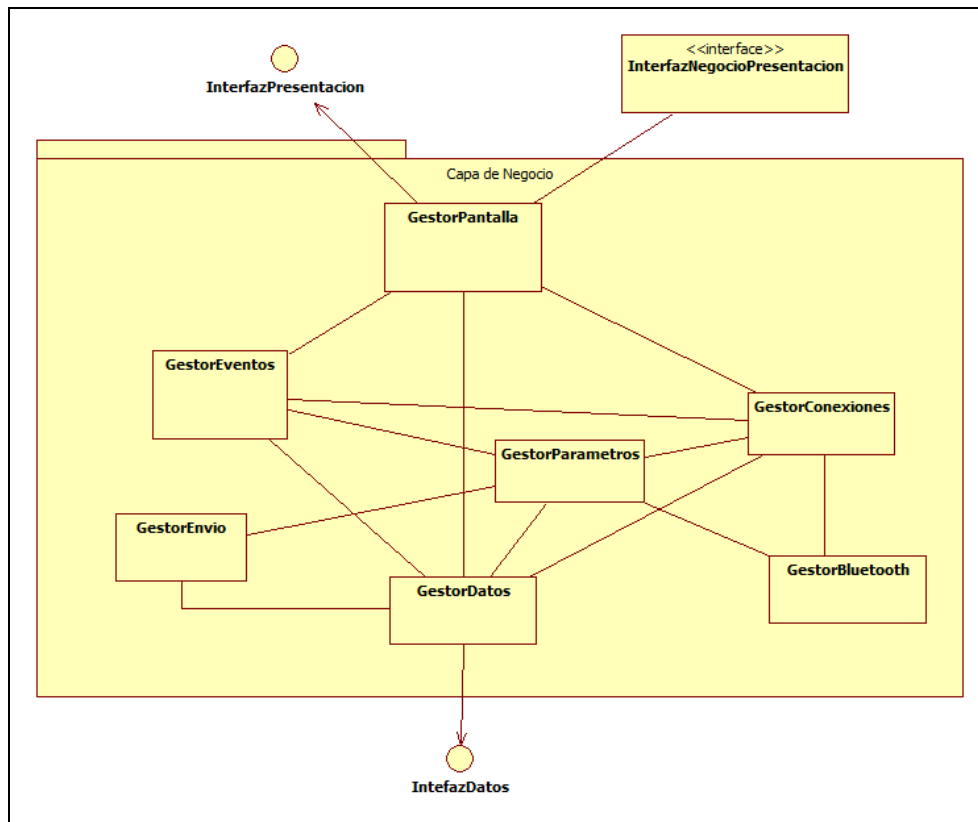


Figura 14: Diagrama de clases Capa de Negocio

En este diagrama se observa la compleja comunicación entre los diferentes componentes de la Capa de Negocio. Para garantizar la claridad de la imagen, se ha suprimido la información interna de las entidades. Cada una de las clases e interfaces contiene una serie de atributos y funciones que se detallan a continuación.

La **interfaz InterfazNegocioPresentacion** permite el acceso a la Capa de Negocio desde la Capa de Presentación. En la siguiente figura se observa la función que ofrece.

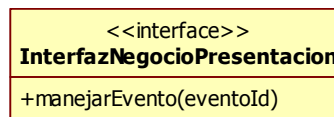


Figura 15: Interfaz InterfazNegocioPresentacion

- **manejarEvento(eventoId)**: transfiere el evento que se ha producido en la pantalla para su gestión en la Capa de Negocio.

La **clase GestorPantalla** es la encargada de establecer la comunicación desde la Capa de Negocio hacia la Capa de Presentación a través de la interfaz InterfazPresentacion. Gestiona todas las funciones que ofrece la interfaz de su capa superior. Esta clase es invocada desde el resto de las entidades de la Capa de Negocio y, por tanto, actúa de intermediario en las operaciones. Además, implementa la función ofrecida por la interfaz InterfazNegocioPresentación. En la siguiente figura se enuncian sus operaciones.

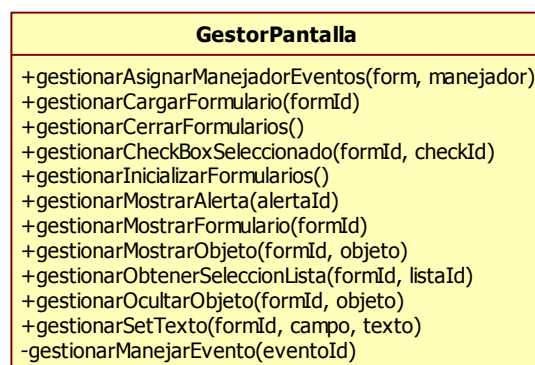


Figura 16: Clase GestorPantalla

Como se comentó anteriormente, estas funciones permiten la comunicación con la Capa de Presentación y, por tanto, su funcionamiento consiste en la preparación de la aplicación para realizar las llamadas a las funciones correspondientes.

La **clase GestorEventos** es la que inicia la ejecución del programa. De igual modo, será la encargada de, por un lado, recoger los eventos generados por la interacción del usuario con la pantalla y, a su vez, procesarlos y por otro, de enviar a la Capa de Presentación la información que debe mostrar. La Figura 17 ilustra la estructura de esta clase.

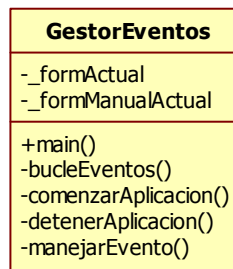


Figura 17: Clase GestorEventos

Los atributos de la clase son privados y son los siguientes:

- **_formActual**: consiste en un valor numérico que almacena el identificador del formulario que está siendo mostrado en pantalla actualmente.
- **_formManualActual**: consiste en un valor numérico que almacena el identificador del formulario del manual que está siendo mostrado en pantalla actualmente.

Por otro lado, las funciones son:

- **comenzarAplicacion()**: realiza las operaciones necesarias para el inicio de la ejecución así como las comprobaciones de los requisitos mínimos del dispositivo.
- **detenerAplicacion()**: realiza las operaciones necesarias para concluir la aplicación.
- **main()**: función pública que permite el inicio de la ejecución del programa.
- **manejarEvento()**: función privada que realiza la gestión de un evento producido en el programa independientemente del origen del mismo.

La **clase GestorConexiones** será la encargada del tratamiento de los sensores, tanto de los acelerómetros como de los RFID. Por tanto, sus funciones serán las de encontrar los dispositivos, conectar con ellos, poner en modo envío de datos, recibir los datos de los sensores y desconectar.

La siguiente figura ilustra los elementos que la componen.

GestorConexiones
- _sensores
+buscarDispositivos() +conectarDispositivos() +conectarSockets() +finalizarCaptura() +finalizarConexiones() +gestionarCerrarBluetooth() +iniciarLectura() +obtenerNombres() +prepararDispositivo() -buscarDispositivoPorSocket(socket) -capturarDatoRFID(sensor, longitudDatos, datos) -capturarDatoWitilt(sensor, longitudDatos, datos) -conectarInputSocket() -formatearDatoRFID(sensor) -formatearDatoWitilt(sensor) -funcionGestionEventosEnlaces(sensor) -hallarTipo(nombre) -inicializarSensores() -obtenerNumeroCreditos(sensor) -sumarNuevosCreditos(sensor, creditos)

Figura 18: Clase GestorConexiones

El único atributo de la clase es:

- **_sensores**: consiste en un vector que almacenará la información relativa de cada sensor. Esta información se define en una estructura que contiene los siguientes elementos: la dirección bluetooth del sensor, el tipo de sensor, un socket de lectura de datos, un socket de escritura de datos, una cadena de caracteres con el último valor leído del sensor.

Asimismo, las funciones son:

- **buscarDispositivos()**: busca los dispositivos que están al alcance del radio Bluetooth y los muestra en una lista para que el usuario elija.
- **conectarDispositivos()**: establece un enlace con el dispositivo que se está tratando en un momento determinado.
- **conectarSocket()**: inicia la conexión de los sockets de entrada mediante la búsqueda del canal de conexión a través del protocolo SDP de Bluetooth.
- **finalizarCaptura()**: realiza las operaciones necesarias tras dar por concluida una ejecución de lectura de datos.
- **finalizarConexiones()**: realiza las operaciones necesarias para cerrar las conexiones con los dispositivos.

- **gestionarCerrarBluetooth()**: realiza las operaciones y llamadas necesarias para la conclusión del uso del Bluetooth.
- **iniciarLectura()**: realiza las operaciones necesarias para comenzar la lectura de datos de los sensores.
- **obtenerNombres()**: obtiene el nombre del dispositivo que se maneja en un momento determinado de la ejecución.
- **prepararDispositivo()**: comprueba si el dispositivo está preparado para usar Bluetooth.
- **buscarDispositivoPorSocket(socket)**: busca la posición del dispositivo en el vector de sensores identificado por el socket de entrada que se recibe por parámetros
- **capturarDatoRFID(sensor, longitudDatos, datos)**: añade en la cadena de datos del sensor RFID cuya posición se pasa por parámetros los datos que ha recibido por el socket de entrada.
- **capturarDatoWiTilt(sensor, longitud, datos)**: añade en la cadena de datos del sensor WiTilt cuya posición se pasa por parámetros los datos que ha recibido por el socket de entrada
- **conectarInputSocket()**: crea la conexión del socket de entrada del sensor que se trata en un momento determinado.
- **formatearDatoRFID(sensor)**: formatea los datos recibidos de un sensor RFID y guarda en la base de datos las lecturas completas de datos de una etiqueta.
- **formatearDatoWiTilt(sensor)**: formatea los datos recibidos de un sensor WiTilt y guarda en la base de datos las cadenas completas de datos.
- **funcionGestionEventosEnlaces(sensor)**: gestiona los eventos producidos por la pila de bluetooth o por los enlaces entre los dispositivos.
- **hallarTipo(nombre)**: halla el tipo del sensor a partir de su nombre bluetooth.
- **inicializarSensores()**:inicializa los valores de los sensores estructurados en el vector de sensores.
- **obtenerNumeroCreditos(sensor)**: obtiene el número de lecturas o créditos que el dispositivo cuya posición en el vector se pasa por parámetros puede realizar mediante el protocolo RFCOMM.
- **sumarNuevosCreditos(sensor, creditos)**: aumenta, la cantidad indicada como parámetro, el número de lecturas que el dispositivo puede realizar mediante RFCOMM.

La **clase GestorBluetooth** es la encargada de trabajar con las funcionalidades Bluetooth del dispositivo. Realiza las operaciones necesarias para establecer las conexiones a través de esta tecnología de comunicación. Para ello, se sirve de las librerías ofrecidas por el lenguaje de programación para dar soporte a las funciones.

Los elementos de la clase son mostrados en la siguiente imagen.

GestorBluetooth
-_libreriaBluetooth
+abrirBluetooth() +cerrarBluetooth() +cerrarSocket() +conectarSocket() +convertirDireccionACadena(direccionBT, cadena) +crearPiconet() +crearSocket(socket, funcion, ref, protocolo) +descubrirDispositivos(texto, boton, filtro, longitudFiltro, numeroDispositivos, nombre, ultimaLista) +desenlazarDispositivo(direccionBT) +desregistrarFuncionNotificacion(funcion) +enlazarDispositivo(direccionBT) +enviarSocket(socket, datos, longitud) +obtenerCanal(socket, direccionBT) +obtenerCreditos(socket, info, creditos, longitud) +obtenerDispositivosSeleccionados(array, numeroDispositivos) +obtenerNombreDispositivo(direccionBT, nombre) +registrarFuncionNotificacion(funcion) +sumarCreditos(socket, creditos)

Figura 19: Clase GestorBluetooth

El único atributo de la clase es:

- **_libreríaBluetooth**: consiste en un valor numérico que representa el identificador de la librería Bluetooth que permite el acceso a las funcionalidades Bluetooth del dispositivo y el manejo de las mismas. Es privado debido a que el acceso a las librerías Bluetooth está restringido únicamente a esta clase.

A continuación, se enumeran las funciones de la clase:

- **abrirBluetooth()**: abre la configuración Bluetooth del dispositivo para ser utilizada. Obtiene el valor de la variable `_libreriaBluetooth`.
- **cerrarBluetooth()**: cierra el uso del Bluetooth del dispositivo y permite que sea utilizado por otras aplicaciones.
- **cerrarSocket(socket)**: cierra el socket identificado por el parámetro recibido
- **conectarSocket(socket, info)**: crea la conexión del socket, cuya referencia se recibe por parámetros, entre el dispositivo local y el remoto a partir de la información indicada. Esta información se compone del canal de conexión así como de la velocidad de transmisión de datos.

- **convertirDireccionACadena(direccionBT, cadena):** convierte la variable de dirección de bluetooth a una cadena de caracteres.
- **crearPiconet():** crea una piconet, cuyo maestro es el dispositivo local, a la que se pueden conectar un máximo de 7 dispositivos.
- **crearSocket(socket, funcion, ref, protocolo):** crea un socket Bluetooth a partir de los parámetros recibidos. Asigna la referencia y el protocolo de conexión indicados y establece la función como la encargada de gestionar los eventos producidos.
- **descubrirDispositivos(texto, boton, filtro, longitudFiltro, numeroDispositivos, nombres, ultimaLista):** busca los dispositivos Bluetooth al alcance del dispositivo. Presenta un listado encabezado por el texto indicado y un botón de aceptación. Busca los elementos que pertenezcan al tipo de dispositivo indicado en el filtro. Permite al usuario seleccionar los que desee. Si el parámetro nombre es verdadero, los dispositivos se identifican por su nombre Bluetooth, en caso contrario lo harán por su dirección Bluetooth. Si el parámetro ultimaLista está activado, se muestra la lista utilizada en la última ejecución. La variable numeroDispositivos almacena el número de dispositivos seleccionados por el usuario.
- **desenlazarDispositivo(direccionBT):** finaliza el enlace con el dispositivo cuya dirección Bluetooth es direccionBT.
- **desregistrarFuncionNotificacion(funcion):** libera la función indicada como la encargada de gestionar los eventos producidos por el bluetooth del dispositivo y los enlaces Bluetooth.
- **enlazarDispositivo(direccionBT):** crea un enlace ACL con el dispositivo cuya dirección Bluetooth es direccionBT.
- **enviarSocket(socket, datos, longitud):** envía por el socket los datos recibidos por parámetros. Además se recibe la longitud de los datos a enviar.
- **obtenerCanal(socket, direccionBT):** obtiene mediante SDP el canal disponible a través del cual puede conectar con el dispositivo, cuya dirección Bluetooth es direccionBT, mediante RFCOMM
- **obtenerCreditos(socket, info, creditos, longitud):** obtiene el número de lecturas que el dispositivo puede realizar mediante RFCOMM. Devuelve el número en la variable créditos.
- **obtenerDispositivosSeleccionados(array, numeroDispositivos):** obtiene un array con las direcciones de los dispositivos seleccionados por el usuario.
- **obtenerNombreDispositivo(direccionBT, nombre):** obtiene el nombre del dispositivo cuya dirección se recibe por parámetros.

- **registrarFuncionNotificacion(funcion):** registra la función indicada por parámetros como la encargada de gestionar los eventos producidos por el estado de Bluetooth del dispositivo y los enlaces Bluetooth que se establezcan.
- **sumarCreditos(socket, creditos):** aumenta el número de lecturas que el dispositivo puede realizar mediante RFCOMM la cantidad indicada por el parámetro créditos.

La **clase Gestor de Datos** es la encargada de la comunicación con la Capa de Datos. Sus funciones serán las de proporcionarle a esta capa los datos que debe escribir así como enviar solicitudes de lectura y recibir los datos leídos.

En la Figura 20 se muestra la estructura de esta clase.

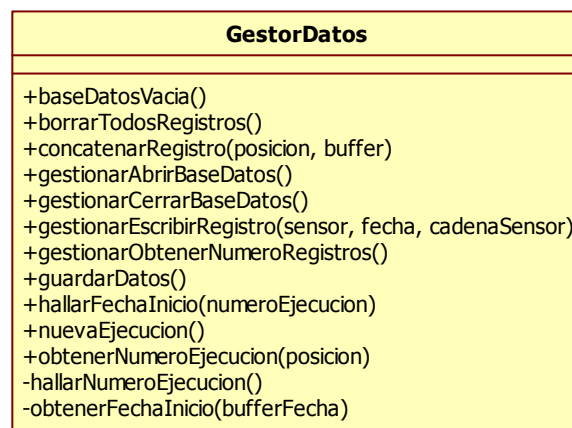


Figura 20: Clase GestorDatos

Las funciones de la clase son las siguientes:

- **baseDatosVacia():** descubre y devuelve si la base de datos está vacía.
- **borrarTodosRegistros():** borra todos los registros de la base de datos
- **concatenarRegistro(posicion, buffer):** copia los campos del registro de la posición indicada en una cadena de caracteres.
- **gestionarAbrirBaseDatos():** proporciona acceso a la función abrirBaseDatos de la Capa de Datos.
- **gestionarCerrarBaseDatos():** proporciona acceso a la función cerrarBaseDatos de la Capa de Datos.
- **gestionarEscribirRegistro(sensor, fecha, cadenaSensor):** proporciona acceso a la función escribirRegistro de la Capa de Datos.

- **gestionarObtenerNumeroRegistros()**: proporciona acceso a la función obtenerNumeroRegistros de la Capa de Datos
- **guardarDatos()**: realiza la transferencia de datos a otro dispositivo mediante Bluetooth
- **hallarFechaInicio(numeroEjecucion)**: halla la fecha de inicio de una ejecución realizada anteriormente.
- **nuevaEjecucion()**: halla el número de ejecución que debe tener la ejecución que va a comenzar.
- **obtenerNumeroEjecucion(posicion)**: obtiene el número de ejecución del registro de la base de datos almacenado en la posición indicada
- **hallarNumeroEjecucion()**: halla el número de ejecuciones realizadas y almacenadas en la base de datos.
- **obtenerFechaInicio(bufferFecha)**: obtiene la fecha actual del sistema y la devuelve formateada en una cadena de caracteres.

La **clase GestorEnvio** es la encargada de realizar la transferencia de datos, almacenados en la base de datos del dispositivo local, a otro dispositivo externo a través de una conexión Bluetooth. Su estructura se muestra a continuación.

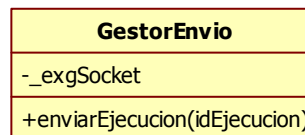


Figura 21: Clase GestorEnvio

El atributo de la clase es el siguiente:

- **_exgSocket**: identificador del socket Bluetooth de intercambio que se establece para la transferencia de datos entre los dispositivos.

La única función de la clase es detallada a continuación:

- **enviarEjecucion(idEjecucion)**: realiza la transferencia de datos de la ejecución indicada almacenados en la base de datos del dispositivo.

La **clase GestorParametros** es una clase auxiliar accedida por la mayoría de elementos de la Capa de Negocio. Contiene una serie de variables que representan la

configuración y el estado del sistema en un momento determinado. La siguiente figura muestra los atributos de esta clase.

GestorParametros
-_capturando -_desplazamientoMilesimas -_frecuencia -_horaInicio -_horaFin -_mostrarValores -_numeroLecturas -_numeroDispositivos -_numeroEjecucion -_proximaLectura

Figura 22: Clase GestorParametros

Los atributos de la clase son detallados a continuación:

- **_capturando:** almacena si los datos proporcionados por los sensores deben ser tratados o no, es decir, si hay una ejecución produciéndose en un momento determinado.
- **_desplazamientoMilesimas:** almacena el número de milésimas de diferencia entre el cambio de segundo y la última vez que se reseteó el dispositivo. Por defecto 0 milésimas.
- **_frecuencia:** almacena la frecuencia de lectura de datos de los sensores.
- **_horaInicio:** almacena la hora de inicio de la ejecución actual.
- **_horaFin:** almacena la hora de finalización de la última ejecución.
- **_mostrarValores:** almacena si los valores que van siendo capturados de los sensores deben ir siendo mostrados por la pantalla en el campo correspondiente.
- **_numeroLecturas:** almacena el número total de lecturas realizadas de los sensores en la ejecución actual.
- **_numeroDispositivos:** almacena el número de dispositivos con los que se establece una conexión en la ejecución actual.
- **_numeroEjecucion:** almacena el número de la ejecución actual dependiendo de las anteriormente guardadas en la base de datos.
- **_proximaLectura:** almacena el instante de la siguiente lectura de los datos de los sensores en un momento determinado.

Las funciones han sido suprimidas de la imagen por motivos de presentación. Por cada variable se han creado dos métodos de acceso. El primero para establecer su valor y el segundo para obtenerlo. Por ejemplo, relacionadas con la variable `_frecuencia`, la función `setFrecuencia(frecuencia)` asigna el nuevo valor a la variable, mientras que `getFrecuencia()` obtiene el valor actual de la misma.

4.3.5 Capa de Datos

Esta última capa es la que se encuentra en el nivel más bajo de la aplicación y su función es la de gestionar los datos que se almacenan en la base de datos del sistema. Por consiguiente, realiza las funciones tanto de lectura como de escritura.

Su composición está formada por una única clase, denominada `BaseDatos`. Por otro lado, esta capa ofrece una interfaz `InterfazDatos` que permite la comunicación desde la capa superior, la Capa de Negocio.

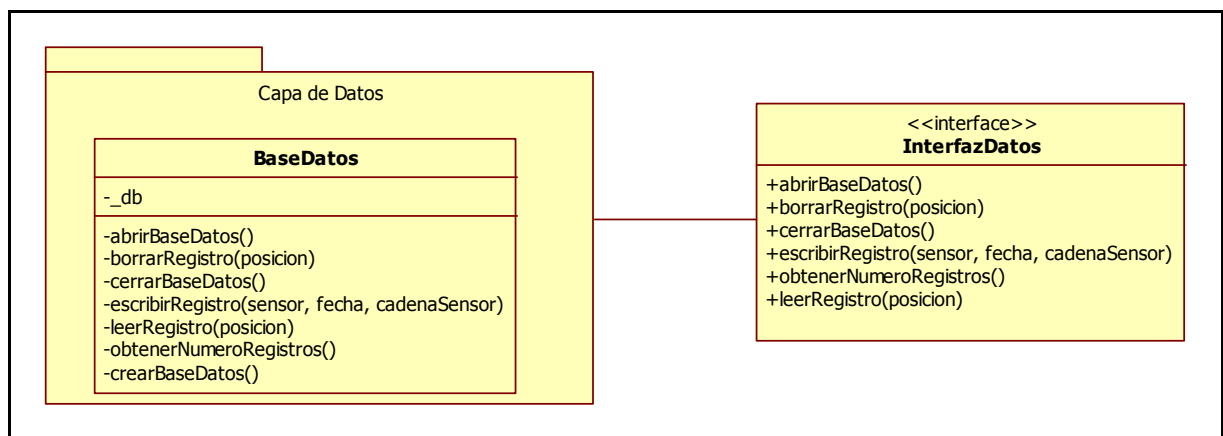


Figura 23: Diagrama de clases Capa Datos

De igual manera que en la Capa de Presentación, la clase implementa las operaciones definidas en la interfaz. Tiene la variable global:

- **_db**: consiste en una referencia al descriptor de la base de datos necesaria para su utilización.

Las funciones realizan operaciones básicas de lectura escritura de la base de datos y son descritas a continuación.

- **abrirBaseDatos()**: abre la base de datos de la aplicación.
- **borrarRegistro(posicion)**: borra de la base de datos el registro de la posición indicada.

- **cerrarBaseDatos():** cierra la base de datos de la aplicación
- **escribirRegistro(sensor, fecha, cadenaSensor):** escribe en la base de datos un registro con los valores indicados.
- **leerRegistro(posicion):** devuelve el registro de la posición indicada.
- **obtenerNumeroRegistros():** obtiene el número de registros almacenados en la base de datos.
- **crearBaseDatos():** crea la base de datos de la aplicación en caso de que sea la primera ejecución en el dispositivo.

4.3.6 Interfaz

El interfaz de la aplicación se generará teniendo en cuenta las restricciones dimensionales del dispositivo central de la aplicación. Sin embargo, todo el proceso de desarrollo debe ser correctamente mostrado por la pantalla para que el usuario advierta el estado de la aplicación en todo momento.

Por esto, cada fase del programa diseñada anteriormente se identificará con claridad y cada una de ellas incluirá solamente la información relevante y necesaria para el usuario.

La interfaz se compondrá de formularios y alertas que se mostrarán dependiendo de la evolución de la aplicación. Sus títulos identificarán de manera clara la fase en la que se encuentra el programa para lograr que el usuario esté informado del proceso en todo momento.

Para comunicar con el usuario se escoge la interacción mediante la pantalla táctil. Para mayor sencillez, se mostrarán una serie de botones que permitirán elegir la acción deseada.

Debido a que en esta aplicación no se requiere la entrada de datos por parte del usuario, no se ha considerado incluir ninguna opción de escritura.

El aprovechamiento del área de la pantalla será proporcional a la cantidad de información a mostrar. Es decir, se centrarán o se ajustarán a los bordes los textos e imágenes dependiendo de la cantidad de ellos que se muestren en cada momento.

Los sensores se representarán en la pantalla mediante un texto con su dirección Bluetooth para identificarlos de una manera cómoda y agradable. Debido al reducido tamaño de la pantalla, sólo si el número de sensores es menor o igual que cuatro se incluirá una imagen adjunta a su dirección.

Como se analizó anteriormente, toda la interacción del usuario con el dispositivo a través de la interfaz será gestionada por la capa de presentación, encargada de la transferencia de las acciones del usuario a la capa de negocio y de la muestra de los resultados del sistema.

Los formularios que se crearán y la relación entre los mismos se muestran en el siguiente diagrama de navegación.

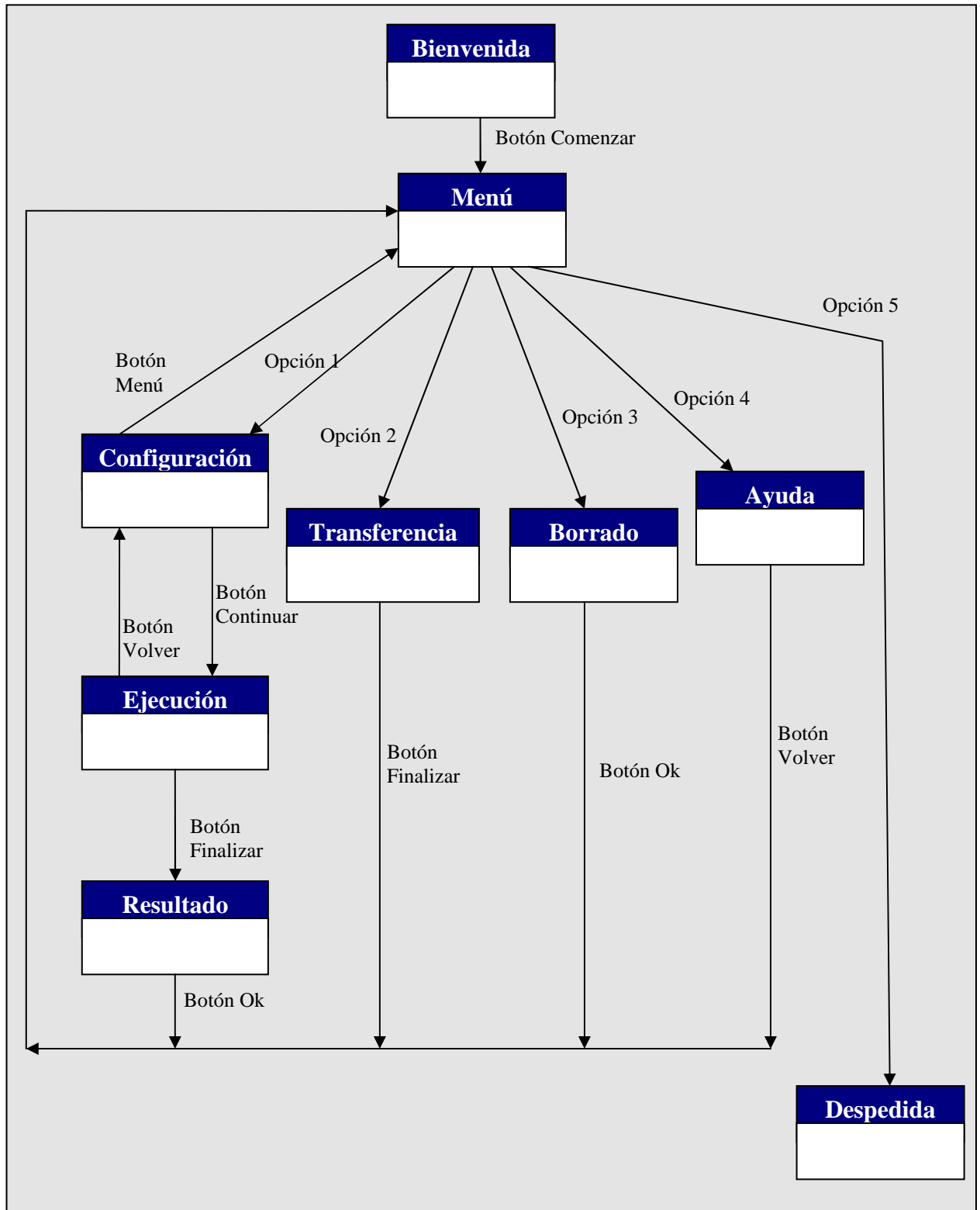


Figura 24: Diagrama de navegación de formularios

Como se puede observar en la anterior figura las transiciones entre los diferentes formularios se realizan tras la selección del usuario bien de un botón o bien de una selección en el menú.

4.3.7 Base de Datos

Con el fin de almacenar los datos obtenidos de las lecturas de los sensores, se han estudiado dos posibilidades. Ambas requieren un posterior tratamiento para la transferencia de los datos a un dispositivo externo. Son estudiadas a continuación:

- **Ficheros de texto:** se puede crear en el dispositivo móvil un fichero de texto por cada ejecución. En él se escribiría cada línea leída de cada sensor. Requeriría la implementación de las operaciones crear, abrir, escribir, leer, cerrar y borrar ficheros. Por otro lado, el proceso de transferencia de los datos a un dispositivo externo implica el acceso y control de los ficheros y la transmisión de cada uno de ellos.
- **Base de datos:** se puede crear en el dispositivo móvil una base de datos en la que se inserten registros para cada lectura realizada por parte de los sensores. Cada registro almacena los datos necesarios para su posterior orden o identificación. Requeriría el control de las funciones típicas con bases de datos como la creación de la base de datos y la inserción, lectura o borrado de un registro. En este caso, en el proceso de transferencia se necesita programar un procedimiento capaz de transformar el contenido de la base de datos en los ficheros finales.

Comparando ambas posibilidades se decantó por la Base de datos debido principalmente a dos criterios. En primer lugar, la escritura en fichero es más lenta que la escritura en base de datos y, teniendo en cuenta que las limitaciones temporales en el programa pueden ser críticas, puede suponer un retardo en la aplicación que influya directamente en su rendimiento. En segundo lugar, la estructura de los ficheros está predefinida de antemano, mientras que el procedimiento de transformación de la base de datos a los ficheros finales introduce la posibilidad de realizar exportaciones con diferentes formatos.

Por todo lo comentado anteriormente, se creará una base de datos en el dispositivo llamada “BDWitilt” que guardará los datos obtenidos en las lecturas de los sensores. Será manejada en la Capa de Datos de la aplicación, concretamente en la clase BaseDatos.

Cada registro está formado por la serie de campos incluidos en la siguiente tabla:

CAMPO	DESCRIPCIÓN	TIPO DE DATO (FORMATO)
ejecucion	El número de ejecución en la que se generó la lectura de datos. Cuando se realiza un vaciado de la base de datos, el número de ejecución se restablece a cero.	Número natural
Sensor	Dirección Bluetooth del sensor del que se ha leído el dato.	Cadena de caracteres de longitud 18 y formato XX:XX:XX:XX:XX:XX
hora	Hora, con precisión de milisegundos, en la que se realizó la captura de datos del sensor	Cadena de caracteres de longitud 12 y formato hh:mm:ss:mmm
cadenaSensor	Datos leídos del sensor	Cadena de caracteres de longitud máxima de 20

Tabla 9: Definición de campos de los registros de la base de datos

A partir de esta definición, se pueden mostrar ejemplos de futuros registros de la base de datos. El primero corresponde con una lectura realizada a un sensor de aceleración, mientras que en el segundo ha sido hecha a un sensor RFID.

Campo	Valor
ejecucion	2
sensor	00:A0:96:1B:30:95
hora	18:58:48:410
cadenaSensor	X=554 Y=576 Z=771

Tabla 10: Ejemplo registro de lectura a un sensor de aceleración

Campo	Valor
ejecucion	3
sensor	00:06:66:00:15:B8
hora	18:59:33:140
cadenaSensor	0105AA93BA

Tabla 11: Ejemplo registro de lectura a un sensor RFID

4.3.8 Ficheros de Salida

En el proceso de transferencia de datos la aplicación establecerá conexión con un dispositivo externo en el cual generará unos ficheros. Estos archivos contendrán toda la información que se ha recibido de los sensores durante las lecturas y se encuentra almacenada en la base de datos de la PDA.

Por cada lectura de datos de los sensores el fichero debe albergar el identificador del sensor, es decir, su dirección Bluetooth, la hora de lectura con la máxima precisión posible (para la Tungsten E2 la máxima precisión es hasta las centésimas) y los datos recibidos del sensor. Además, se debe conocer la hora de inicio de la ejecución.

Para elaborar el diseño del formato de los ficheros se ha tenido en cuenta que para mejorar el rendimiento del proceso de transferencia se debe reducir, en la medida de lo posible, la cantidad de información que se almacena y el procesamiento de los datos.

Tras varias propuestas de formatos estudiados, el elegido tiene las siguientes características:

- Cada ejecución completa de la conexión con los sensores tiene asignada un identificador numérico que se almacena en el campo ejecución de los registros de la base de datos. Para cada valor de este registro se va a generar un único fichero. Su nombre será la fecha de inicio de la ejecución con sus elementos separados por el carácter “_”. Su extensión será “.txt” que es la típica en ficheros de texto. Un ejemplo de nombre de fichero es 2009_02_24_12_17_59.txt. La siguiente figura ilustra este proceso de creación de ficheros.

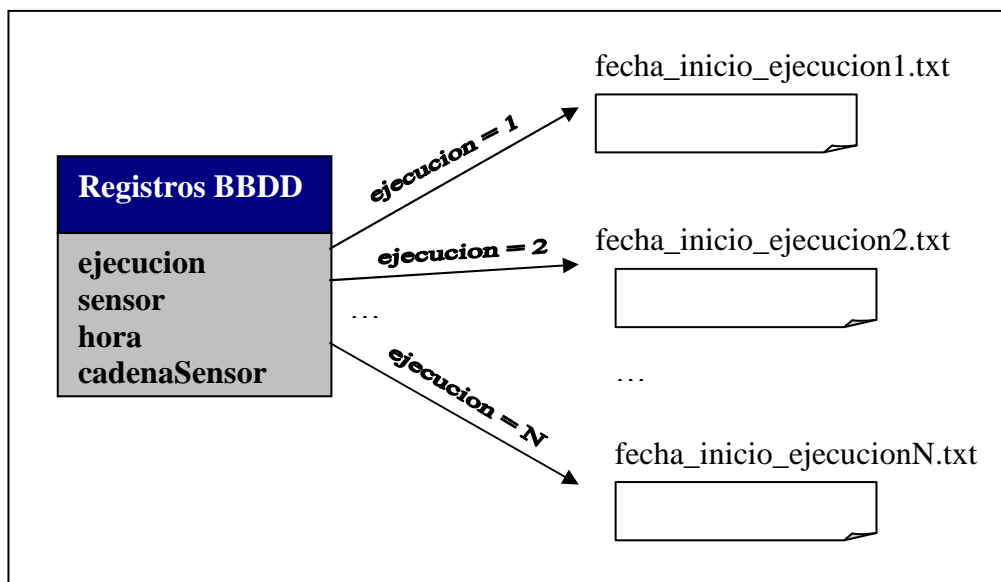
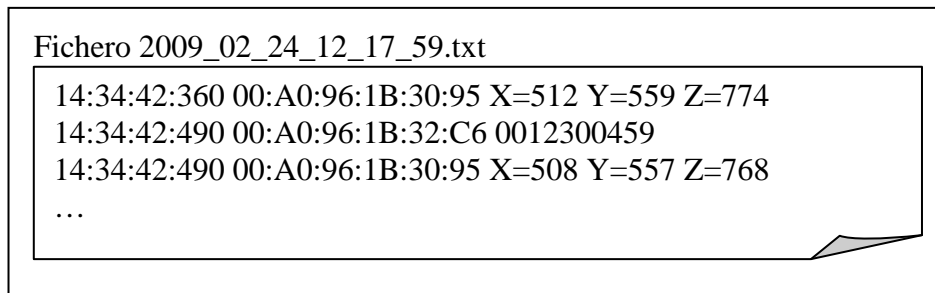


Figura 25: Creación de ficheros de transferencia

- El contenido del fichero estará formado por una línea por lectura. Cada línea contendrá la cadena con la hora de captura con sus elementos separados por el carácter “:”, la dirección Bluetooth del sensor y la cadena de datos leídos. Todos los campos están separados por espacios. Un ejemplo del contenido de un fichero es el mostrado en la siguiente figura.

El diagrama muestra un archivo de texto con el nombre 'Fichero 2009_02_24_12_17_59.txt'. Dentro del archivo, se listan tres líneas de datos de sensores, cada una con una hora de captura, una dirección Bluetooth y tres valores de coordenadas (X, Y, Z). Las líneas de datos son: '14:34:42:360 00:A0:96:1B:30:95 X=512 Y=559 Z=774', '14:34:42:490 00:A0:96:1B:32:C6 0012300459' y '14:34:42:490 00:A0:96:1B:30:95 X=508 Y=557 Z=768'. Después de la tercera línea, hay tres puntos de suspensión '...' para indicar que el archivo contiene más datos. El diagrama está representado como una ventana de texto con una esquina inferior derecha doblada.

```
Fichero 2009_02_24_12_17_59.txt
14:34:42:360 00:A0:96:1B:30:95 X=512 Y=559 Z=774
14:34:42:490 00:A0:96:1B:32:C6 0012300459
14:34:42:490 00:A0:96:1B:30:95 X=508 Y=557 Z=768
...
```

Figura 26: Ejemplo de fichero de transferencia

4.4 Implementación de la Aplicación

En este apartado se describe cómo se ha realizado el proceso de implementación de la aplicación final teniendo en cuenta los estudios realizados en todas las fases anteriores.

4.4.1 Configuración del Desarrollo

Para la implementación del diseño del proyecto se ha utilizado el entorno de programación de aplicaciones para Palm denominado Garnet OS Development Suite [32]. Incluye tanto los programas como las librerías necesarios para la edición, compilación, simulación y depuración de una aplicación realizada para dispositivos con sistema operativo PalmOS. Está formado por los siguientes programas:

- El programa de desarrollo de aplicaciones Garnet OS IDE versión 1.2.1.02. Está basado en la tecnología del proyecto Eclipse [33] e incluye las librerías necesarias para la programación. Permite la escritura del código fuente y la posterior compilación para la creación de ficheros ejecutables en dispositivos Palm.
- El programa de diseño de interfaces Palm OS resource editor versión 6.2.0.6. Incluye software desarrollado por la fundación Apache [34]. Consiste en un editor gráfico de interfaces, en el cual se generan los formularios que serán mostrados por la pantalla en el programa. Asigna a cada elemento gráfico un

identificador a través del cual se accederá a ellos desde el código fuente de la aplicación.

- El programa de simulación Palm OS Emulator 3.5. Es utilizado para simular la ejecución de aplicaciones en una plataforma Palm OS. Consiste en una ventana con la apariencia de un dispositivo Palm que permite la ejecución de ficheros ejecutables para el sistema operativo PalmOS.

Empleando el entorno de desarrollo definido arriba, la metodología de trabajo que ha marcado el desarrollo de la aplicación contiene los siguientes pasos:

1. Creación de un nuevo proyecto en el programa de desarrollo de aplicaciones.
2. Creación del fichero con extensión “xrd”. Apertura del fichero en el programa de diseño de interfaces.
3. Creación de los formularios y su contenido de la interfaz de usuario, identificando cada elemento con un número.
4. Importación de la interfaz creada en el fichero con extensión “xrd”.
5. Creación del fichero de identificación de recursos. En él se crean constantes a las que se asignan nombres a los identificadores de cada uno de los elementos de la interfaz.
6. Creación del código fuente con el programa Garnet OS Development Suite (ficheros .c y .h) apoyándose en las librerías definidas para Palm.
7. Compilación automática y posterior creación automática del fichero ejecutable (extensión “prc”).
8. Simulación de la ejecución en el emulador.
9. Sincronización entre el ordenador y el dispositivo (HotSync) para la instalación del ejecutable en la PDA.
10. Ejecución de la aplicación en la PDA.

4.4.2 Interfaz de Usuario

A continuación se incluyen imágenes donde se muestra la apariencia gráfica de los formularios creados que componen la interfaz de usuario.

Como se observa en el diagrama de navegación de formularios de la Figura 24, el formulario de bienvenida es el primero que se muestra al iniciarse el programa. Incluye el botón “Comenzar” para entrar en la aplicación.



Figura 27: Formulario de bienvenida

Inmediatamente después se muestra el formulario con el menú. Incluye las distintas opciones que tiene un usuario en la aplicación. Dichas opciones son seleccionadas pulsando directamente sobre ellas. En la siguiente figura se aprecia la apariencia de este formulario.



Figura 28: Formulario Menú

La primera opción del menú conduce al formulario de configuración. Éste permite al usuario seleccionar dos parámetros para la ejecución de las lecturas de datos. El primero es el número de sensores con los que se va a establecer la comunicación y el segundo la frecuencia de recogida de los datos enviados por los dispositivos. Ambos se seleccionan mediante la elección de valores predeterminados insertados en sendas listas desplegables. El botón “Continuar” permite avanzar en el programa, mientras que el botón “Menu” regresa al anterior formulario.

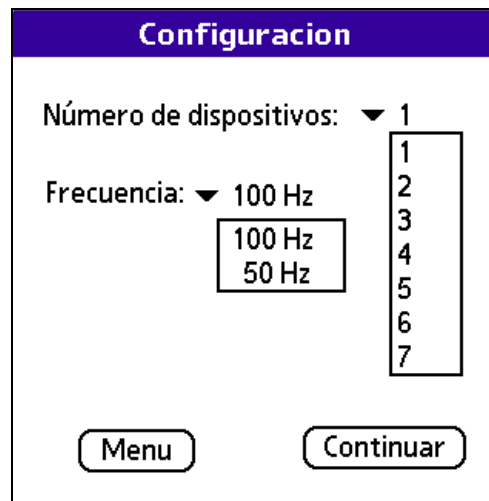


Figura 29: Formulario Configuración

Si el botón “Continuar” del formulario de configuración es pulsado aparece en pantalla el de ejecución. En él se muestra información acerca de los sensores con los que se trata de conectar. Si el número de dispositivos es menor o igual que cuatro se incluye una imagen que los representa. Además, se escribe la dirección Bluetooth de cada uno y los datos que se reciben a través de las conexiones.

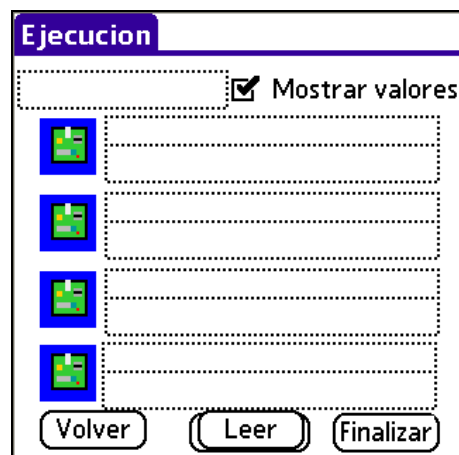
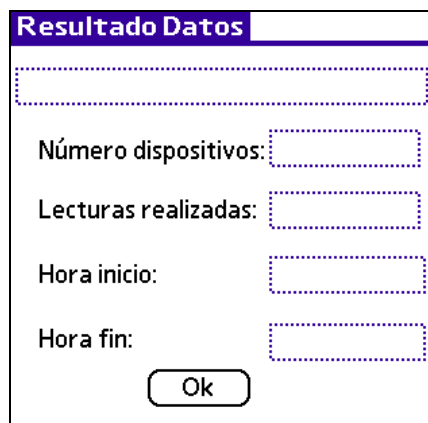
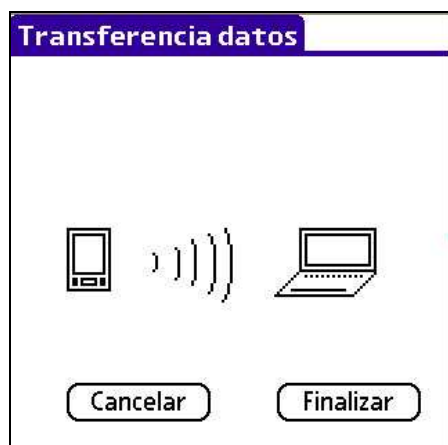


Figura 30: Formulario Ejecución

Al seleccionar el botón “Finalizar” del formulario ejecución se muestra el de resultados. Éste último incluye la información relativa a la ejecución llevada a cabo y que acaba de concluir. Cuando se pulsa el botón “Ok”, el programa vuelve a su menú principal. Su imagen es la siguiente.

A screenshot of a software window titled "Resultado Datos". It features a purple header bar. Below the header is a large empty rectangular box. Underneath this box are four labels with corresponding input fields: "Número dispositivos:", "Lecturas realizadas:", "Hora inicio:", and "Hora fin:". Each label is followed by a dotted rectangular input field. At the bottom center of the window is an "Ok" button.**Figura 31: Formulario Resultado**

La opción 2 del menú principal conduce al formulario de transferencia de datos a un fichero externo. A partir de la imagen que se muestra abajo, se ofrece el listado de dispositivos en el que el usuario debe elegir el destinatario de los datos. Si selecciona el botón "Cancelar" se retrocede al menú de la aplicación cancelando todos los envíos. Si se selecciona el botón "Finalizar" se cancela el envío actual.

A screenshot of a software window titled "Transferencia datos". It has a purple header bar. The main area contains an icon of a mobile phone on the left and a laptop on the right, connected by four curved lines representing data transfer. Below these icons are two buttons: "Cancelar" on the left and "Finalizar" on the right.**Figura 32: Formulario Transferencia**

Si en el menú principal se selecciona la opción 3, se muestra por pantalla el formulario de borrado de datos. Debido a que este proceso provoca la pérdida irreversible de los datos se ofrece una pregunta de confirmación al usuario para evitar el borrado indeseado. Si se selecciona el botón "No" se cancela la operación y se vuelve al menú principal del programa. Sin embargo, si por el contrario se selecciona el botón "Si" se procede al borrado. A su término se muestra el botón "Ok" que permite la vuelta al menú principal del programa.

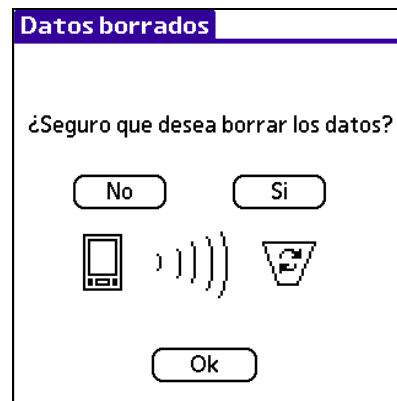


Figura 33: Formulario Borrado

Al manual de usuario se accede a través de la opción 4 del menú principal de la aplicación. Está compuesto por una serie de formularios que muestran información relativa a las tareas más importantes de la ejecución y que pueden llevar a errores. Poseen botones que permiten la navegación entre ellos. El primer formulario es el que se muestra a continuación.

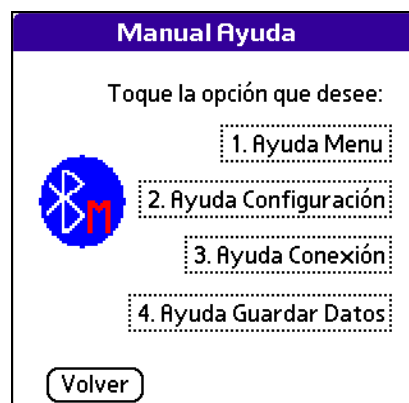


Figura 34: Formulario Ayuda

Como se comentó en el apartado anterior, todos los elementos introducidos en la interfaz, como son los formularios, las alertas, los botones o los textos, son definidos, junto con sus identificadores, en un fichero de texto, denominado `bluemovedRsc.h`, para poder ser manipulados por el código fuente del programa.

4.4.3 Código Fuente

A la hora de implementar el diseño de la aplicación previamente realizado, se han generado ficheros con el código fuente que, tras su compilación, formarán el fichero ejecutable. Para ello, se ha empleado el lenguaje de programación C.

La totalidad de los archivos que contienen el código fuente de la aplicación tienen una misma estructura y nomenclatura. En la parte superior se incluye información acerca de la función de cada fichero en la aplicación global y el elemento que representan del diseño. A continuación se muestran las relaciones con otros ficheros que son necesarias para su funcionamiento. Posteriormente, se enumeran las librerías que han sido incluidas para permitir el acceso a los métodos ya definidos en el lenguaje de programación. Seguidamente, se muestran las constantes y las variables globales creadas para cada fichero. Finalmente, se escriben las funciones que van a llevar a cabo el control del flujo de la ejecución del programa.

Asimismo, para facilitar la comprensión al lector, el código ha sido debidamente comentado. Se han explicado tanto la labor de cada fichero como el uso de cada variable y función, así como los detalles de funcionamiento de prácticamente cada línea del programa.

La siguiente tabla enumera los ficheros de código creados y los elementos del diseño que representa cada una de ellas.

Fichero Código Fuente	Entidad Representada del Diseño
basedatos.c	Clase BaseDatos
basedatos.h	Interfaz InterfazDatos
beaming.c	Clase GestorEnvio
beaming.h	
bluemoved.c	Clase GestorEventos Interfaz InterfazNegocioPresentacion
bluetooth.c	Clase GestorBluetooth
bluetooth.h	
conexion.c	Clase GestorConexion
conexion.h	
gestorDatos.c	Clase GestorDatos
gestorDatos.h	
gestorPantalla.c	Clase GestorPantalla
gestorPantalla.h	
pantalla.c	Clase Presentacion
pantalla.h	Interfaz InterfazPresentacion
parametros.c	Clase GestorParametros
parametros.h	

Tabla 12: Ficheros con código fuente de la aplicación

4.4.4 Descripción del Funcionamiento de la Aplicación

En este apartado se realiza una explicación detallada de la implementación de las tareas más complejas que realiza la aplicación.

- **Arranque y terminación de la aplicación:**

Como se comenta en la sección anterior, la ejecución comienza en el fichero blumoved.c y, por tanto, es el que contiene la función *main* de la aplicación.

La primera acción a realizar es verificar que el dispositivo cumple los requisitos mínimos de la aplicación (versión del Sistema Operativo PalmOS 2.0 o superior). A continuación, se comprueba que está activo tanto el control del Bluetooth del terminal como la gestión de la base de datos. Todas estas operaciones de control se realizan en la función *comenzarAplicacion()* definida en la clase *GestorEventos* del diseño. Si de todo lo anterior se obtiene una respuesta afirmativa, el programa carga el primer formulario en la pantalla y entra en un bucle.

Aparte del arranque, este fichero es el encargado de manejar los eventos producidos en la aplicación y llevar el control del flujo del programa. Para ello, se genera un bucle que recibe los eventos producidos a través de una pila de eventos. Este bucle se ejecuta mientras que no se produzca un evento de finalización y se encuentra definido en la función *bucleEventos()*.

Cuando el bucle termine, debido a que se ha producido el evento de fin de ejecución, se realizan las operaciones necesarias para la conclusión de la ejecución. Estas son cerrar la conexión Bluetooth del dispositivo, cerrar la Base de Datos y cerrar los formularios. Todas ellas se realizan en la función *detenerAplicacion()*.

- **Establecimiento de las conexiones Bluetooth:**

A la hora de establecer las conexiones con los dispositivos Bluetooth, se crean sockets que permiten las comunicaciones dentro de la piconet creada en la aplicación. Las acciones realizadas se encuentran programadas en los ficheros *conexion.c* y *bluetooth.c*.

Tras realizar la búsqueda de los dispositivos y obtener la lista seleccionada por el usuario, se recorre esta lista para establecer los enlaces a través de la función *enlazarDispositivo(direccionBT)*. Se realiza de manera secuencial, es decir, no se trata de conectar con un sensor hasta que se haya establecido la conexión con el anterior. De este modo, se comprueba en todo momento, si alguno de los enlaces no se produce correctamente, lo que detendría la ejecución.

Si los enlaces son establecidos correctamente, se generan los sockets RFCOMM de comunicación, utilizando las funciones *crearSocket()* y *conectarSocket()*. De igual modo que el anterior proceso, se realiza secuencialmente para la comprobación de errores.

En el caso en el que todo sucede correctamente, cuando el usuario seleccione el botón “Leer” del formulario “Formulario Ejecucion” se ejecuta la función *iniciarLectura()*. Esta envía, a través de los sockets, el parámetro de configuración a cada sensor, empleando la función *enviarSocket(socket, datos, longitud)*. De esta manera, los sensores se ponen en modo de envío de datos, e inmediatamente comienza la lectura de datos en el dispositivo.

Cuando el usuario, selecciona el botón “Finalizar” del formulario “Formulario Ejecucion”, las lecturas concluyen empleando las funciones finalizarCaptura() y finalizarConexiones() que a su vez cierran las conexiones, cerrarSocket(socket).

- **Tratamiento de los datos de los sensores:**

Dependiendo del tipo de sensor, los datos obtenidos siguen un formato determinado. Los sensores de aceleración envían cadenas de caracteres que contienen información sobre los tres ejes de coordenadas, mientras que los RFID envían cadenas que conforman un identificador de etiqueta.

En las lecturas realizadas por los sockets se reciben un número variable de caracteres. La composición de una cadena completa de datos, para su posterior inserción en la base de datos, puede necesitar la ejecución de más de una lectura debido a que la cadena completa puede haber sido dividida para su envío. Los datos recibidos en las lecturas son analizados en las funciones de captura y formateo de datos, que son capturarDatoWitilt(), capturarDatoRFID(), formatearDatoWitilt() y formatearDatoRFID(). Las dos primeras concatenan las cadenas que se reciben del sensor en un momento dado con el final de las recibidas en lecturas previas. En cambio, las dos últimas comprueban el contenido de las cadenas, obteniendo trozos completos y realizando los almacenamientos de los mismos.

Para la lectura y escritura de los datos en la base de datos se utilizan las funciones del fichero basedatos.c leerRegistro y escribirRegistro a las cuales se puede acceder desde la Capa de Negocio a través del fichero gestorDatos.c y sus funciones gestionarLeerRegistro y gestionarEscribirRegistro().

- **Transferencia de datos a dispositivo externo:**

La transferencia de los datos obtenidos de las diferentes ejecuciones es implementada en el fichero beaming.c y, en concreto, en la función enviarEjecucion(). Sin embargo, el proceso lógico se realiza en el fichero gestorDatos. Como los envíos se realizan por ejecuciones, se halla el número de ejecuciones que actualmente se encuentran almacenadas en la base de datos mediante la función hallarNumeroEjecucion(). Se realizan tantos envíos como ejecuciones halla y, para ello, con cada ejecución se recorre la base de datos buscando los registros que pertenezcan a la misma. Esto se comprueba comparando el campo “ejecucion” de cada registro.

Para realizar el envío, se genera un socket de comunicación directa entre ambos dispositivos. Se establece el nombre del fichero a partir de la fecha en la que se inició la lectura. Se abre y se comienza a escribir todos los registros obtenidos de la base de datos, pertenecientes a la ejecución a enviar, cada uno en una línea del fichero. Al término, se cierra el socket.

5. Evaluación de la Aplicación

En este capítulo se describe el proceso de evaluación que se ha llevado a cabo sobre la aplicación final.

La primera tarea consiste en redactar las principales pruebas que se han definido para evaluar tanto la funcionalidad de la aplicación como su calidad y rendimiento. Seguidamente, se diseñan unas tablas que sirven para la recogida y muestra de los datos obtenidos de las diferentes pruebas llevadas a cabo. A continuación, se describe la realización de los diferentes tests y los resultados de cada una de ellas. Finalmente, se redactan las conclusiones extraídas a partir de los resultados de las pruebas y se analiza una completa evaluación de la aplicación final.

5.1 Definición Pruebas de Evaluación

Para realizar la evaluación de la aplicación se han diseñado una serie de pruebas que son llevadas a cabo mediante ejecuciones reales del programa en el dispositivo. Estos tests son divididos en dos grandes grupos dependiendo de su objetivo.

En primer lugar se definen las pruebas de funcionamiento. Consisten en la comprobación de que la aplicación cumple los requisitos de funcionalidad programados en los objetivos del programa. En resumen, contrastan si los principales procesos son llevados a cabo correctamente. Estos son el establecimiento de los enlaces con los sensores, la conexión con los mismos y las lecturas de datos.

Para esta fase, se han definido las siguientes evaluaciones:

- Conexión con un sensor de aceleración
- Conexión con un dispositivo RFID
- Conexión con varios sensores de aceleración
- Conexión con varios sensores RFID
- Conexión con varios sensores de ambos tipos.

Conjuntamente con las anteriores, se especifican una serie de pruebas para comprobar el correcto funcionamiento del proceso de transferencia de datos a un dispositivo externo. Estas son:

- Transferencia de una ejecución
- Transferencia de varias ejecuciones
- Transferencia de varias ejecuciones tras salir del programa

Del mismo modo, se idean unas pruebas de calidad en las que se verifican los datos almacenados en el dispositivo. Es decir, tras la transferencia de los datos, éstos son chequeados en los ficheros creados en busca de posibles errores.

- Comprobación de datos con un sensor de aceleración.
- Comprobación de datos con un sensor RFID.
- Comprobación de datos con varios sensores.

Una vez realizados tanto los tests de funcionamiento como los de calidad, se llevan a cabo los de rendimiento. El objetivo de esta fase es la de evaluar de manera cuantitativa el funcionamiento del programa. En concreto, consiste en contrastar la frecuencia de recogida de datos real con la estipulada en la configuración de la aplicación. Se tratará de comprobar cómo influyen en ella aspectos tales como el número de conexiones o la frecuencia de configuración de envío de los sensores.

Para efectuar este apartado de la evaluación, se han definido las siguientes pruebas:

- Conexión con un sensor de aceleración
- Conexión con varios sensores de aceleración
- Conexión con un sensor RFID
- Conexión con todos los sensores disponibles
- Estudio de almacenamiento

5.2 Definición Tablas de Resultados de Evaluación

En este apartado se describe la estructura de las tablas que van a servir para mostrar la recogida de datos obtenidos a partir de las diferentes pruebas realizadas.

Como se indicó anteriormente, las evaluaciones se dividen en tres grupos, las de funcionamiento, las de calidad y las de rendimiento, para los cuales se usarán las cadenas “EVF”, “EVC” y “EVR” respectivamente. En el identificador de cada tabla se añadirá un número de cuatro cifras que indica el orden en que fueron realizadas. Además se describirá brevemente las conexiones y objetivos de cada proceso.

En primer lugar, se muestra la tabla de datos de los test de funcionamiento de las conexiones.

Evaluación Id	Descripción			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos

Tabla 13: Plantilla de test de funcionamiento de conexiones

Como se puede observar, en ella se incluyen datos acerca de las conexiones establecidas, escribiéndose una línea por cada uno de los sensores con los que se conecten. Por un lado, información sobre el sensor como es su dirección y su tipo. Seguidamente, información sobre el resultado de los procesos necesarios para la correcta ejecución del programa, como son el enlace con el sensor, la conexión y la lectura de datos. En estos últimos campos se insertará la cadena “Correcto”, si se realizó correctamente, o “Error” si se produjo algún fallo.

Las siguientes pruebas que se realizan son los denominados test de funcionamiento de las transferencias. Para su registro se utilizará la siguiente tabla.

Evaluación Id	Descripción			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero

Tabla 14: Plantilla de test de funcionamiento de transferencia

Para su construcción se ha añadido información acerca del proceso de transferencia de datos a la tabla definida para el funcionamiento de las conexiones. La nueva línea se compone de cuatro campos: el dispositivo al que se realiza la transferencia, el resultado del proceso, el nombre del fichero creado y el tamaño del mismo. Si se realizan varias transferencias debido a que existía más de una ejecución almacenada en la base de datos del dispositivo, se incluye una línea para cada una de ellas.

A la hora de registrar los resultados obtenidos en los test de calidad, se mostrarán los datos en una tabla con el siguiente formato.

Evaluación Id	Descripción			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Configuración				
Frecuencia lectura de datos		Frecuencia Configuración Sensor		
Datos				
Número Datos	Datos Incorrectos	Porcentaje de Datos Incorrectos		

Tabla 15: Plantilla de test de calidad

En cuanto a las pruebas de rendimiento de la aplicación, es necesario incluir información acerca de los datos transferidos al dispositivo de destino. Primeramente, se escribe información sobre la frecuencia de lectura de datos establecida en el sensor para el envío y, a partir de ella, el número de datos estimado que se debería obtener. Además se incluye la frecuencia de configuración de lectura de datos del programa. Posteriormente, se analiza el contenido del fichero indicando el número real de datos y el porcentaje respecto a los estimados. Para finalizar, se calcula la frecuencia de datos obtenida a partir de la división del número de lecturas realizadas entre la duración. En última instancia se muestra el nombre del fichero donde se recogen los datos expuestos. Esta información se presenta en la siguiente tabla.

Evaluación Id		Descripción				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero

Tabla 16: Plantilla de test de rendimiento

5.3 Realización Pruebas de Evaluación

En este apartado se recogen los datos de las ejecuciones reales de las diferentes ejecuciones realizadas en esta fase de evaluación. La información se muestra en las tablas creadas para tal fin.

5.3.1 Pruebas de Evaluación del Funcionamiento

Se comienza con las pruebas de evaluación definidas para comprobar el correcto funcionamiento de la aplicación.

- **Conexión con un sensor de aceleración:**

Consiste en la prueba básica a realizar con los sensores de aceleración. En la siguiente tabla se puede observar cómo las tres fases que componen la comunicación con el sensor, enlace, conexión y lectura de datos, han obtenido un resultado correcto.

Evaluación Id	Descripción			
EVF-0001	Conexión con un sensor de aceleración para comprobar el correcto funcionamiento de la aplicación.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:30:D0	Aceleración	Correcto	Correcto	Correcto

Tabla 17: Test de funcionamiento con un sensor de aceleración

- **Conexión con un dispositivo RFID:**

Al igual que el anterior test, consiste en la prueba básica a realizar, pero en este caso, con un sensor RFID. Como en la prueba anterior, los tres procesos se han llevado a cabo correctamente y la información se ha recogido en la siguiente tabla.

Evaluación Id	Descripción			
EVF-0002	Conexión con un sensor RFID para comprobar el correcto funcionamiento de la aplicación.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:06:66:00:15:B8	RFID	Correcto	Correcto	Correcto

Tabla 18: Test de funcionamiento con un sensor RFID

- **Conexión con varios sensores de aceleración:**

En este test la complejidad aumenta puesto que es necesario establecer comunicación con tres sensores simultáneamente. Como se puede observar en la siguiente tabla, el resultado es el esperado y todos los procesos provocan una comunicación satisfactoria.

Evaluación Id	Descripción			
EVF-0003	Conexión simultánea con tres sensores de aceleración para comprobar el correcto funcionamiento de la aplicación.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:30:D0	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:32:C6	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:30:95	Aceleración	Correcto	Correcto	Correcto

Tabla 19: Test de funcionamiento con tres sensores de aceleración

-
- **Conexión con varios sensores RFID:**

No se dispone de más de un sensor RFID y, por tanto, no puede realizarse esta prueba.

- **Conexión con varios sensores de ambos tipos:**

La dificultad de esta prueba radica en que el sistema debe establecer comunicación con varios sensores independientemente del tipo que sea cada uno y de manera transparente al usuario. Los resultados obtenidos, y resumidos en la siguiente tabla, muestran que todos los procesos producen un resultado correcto y, por tanto, se obtienen datos de todos los sensores.

Evaluación Id	Descripción			
EVF-0004	Conexión simultánea con tres sensores de aceleración y uno RFID para comprobar el correcto funcionamiento de la aplicación.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:30:D0	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:32:C6	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:30:95	Aceleración	Correcto	Correcto	Correcto
00:06:66:00:15:B8	RFID	Correcto	Correcto	Correcto

Tabla 20: Test de funcionamiento con tres sensores de aceleración y uno RFID

- **Transferencia de una ejecución:**

En esta prueba se comprueba si el proceso de transferencia de datos a otro dispositivo externo se realiza correctamente. Para ello, primero se realiza una ejecución como cualquiera de las anteriores y a su finalización se escoge la opción de transferir datos. En este caso concreto, se ha partido de la evaluación EVF-0004 mostrada previamente.

Evaluación Id	Descripción			
EVF-0005	Transferencia a un dispositivo externo de los datos recogidos en la ejecución de evaluación EVF-0004.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:30:D0	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:32:C6	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:30:95	Aceleración	Correcto	Correcto	Correcto
00:06:66:00:15:B8	RFID	Correcto	Correcto	Correcto
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Ordenador portátil	Correcto	2009_04_17_13_41_45.txt		9 KB

Tabla 21: Test de funcionamiento transferencia de una ejecución

Como se puede observar en la anterior tabla, el resultado de la transferencia es correcto y el fichero generado tiene el nombre “2009_04_17_13_41_45.txt” y un tamaño de 9 KB. Un trozo de este archivo creado en el ordenador portátil es el mostrado en la siguiente figura.

13:41:46:380	00:A0:96:1B:32:C6	X=525	Y=493	Z=780
13:41:46:400	00:A0:96:1B:30:95	X=519	Y=542	Z=772
13:41:46:420	00:A0:96:1B:32:C6	X=524	Y=494	Z=783
13:41:46:480	00:A0:96:1B:30:D0	X=524	Y=482	Z=768
13:41:46:500	00:A0:96:1B:32:C6	X=523	Y=492	Z=782
13:41:46:520	00:A0:96:1B:30:D0	X=523	Y=486	Z=769
13:41:46:540	00:A0:96:1B:32:C6	X=525	Y=496	Z=780
13:41:46:560	00:A0:96:1B:30:95	X=525	Y=539	Z=771
13:41:46:580	00:A0:96:1B:30:95	X=515	Y=542	Z=770
13:41:46:600	00:A0:96:1B:30:95	X=520	Y=540	Z=775
13:41:46:620	00:A0:96:1B:30:D0	X=521	Y=483	Z=771
13:41:46:640	00:A0:96:1B:30:95	X=520	Y=544	Z=775
13:41:46:660	00:A0:96:1B:30:D0	X=523	Y=481	Z=771
13:41:46:680	00:A0:96:1B:30:D0	X=522	Y=486	Z=768
13:41:46:710	00:06:66:00:15:B8	0105AA87D4		
13:41:46:730	00:06:66:00:15:B8	0105AA87D4		
13:41:46:750	00:A0:96:1B:32:C6	X=526	Y=494	Z=780
13:41:46:770	00:A0:96:1B:30:95	X=519	Y=546	Z=775
13:41:46:790	00:A0:96:1B:30:95	X=518	Y=543	Z=771
13:41:46:810	00:A0:96:1B:30:D0	X=522	Y=482	Z=768

Figura 35: Porción de código del fichero creado en la evaluación de transferencia

-
- **Transferencia de varias ejecuciones:**

Este test se realiza para comprobar la posibilidad de transferir al dispositivo externo varias ejecuciones al mismo tiempo. Con este objetivo se han llevado a cabo tres ejecuciones de lectura con un sensor de aceleración y, al término de la tercera, se ha realizado la transferencia de datos.

Evaluación Id	Descripción			
EVF-0006	Transferencia a un dispositivo externo de tres ejecuciones distintas al mismo tiempo.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:32:C6	Aceleración	Correcto	Correcto	Correcto
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Ordenador portátil	Correcto	2009_04_17_13_52_23.txt		10 KB
Ordenador portátil	Correcto	2009_04_17_13_53_21.txt		7 KB
Ordenador portátil	Correcto	2009_04_17_13_53_51.txt		5 KB

Tabla 22: Test de funcionamiento transferencia de tres ejecuciones

Como se puede observar en la tabla anterior, se han generado tres ficheros con los datos de cada una de las ejecuciones.

- **Transferencia de ejecuciones tras apagado del dispositivo:**

La siguiente prueba se realiza para demostrar que el programa es capaz de transferir los datos de antiguas ejecuciones incluso después del apagado del dispositivo. Para ello, se ha procedido de igual manera que en el anterior test pero con la particularidad del apagado y posterior encendido de la PDA. Por tanto, se han realizado tres ejecuciones de comunicación con un sensor, al término de la tercera se ha desconectado la PDA, posteriormente ha vuelto a ser encendida y finalmente se ha realizado la transferencia.

Evaluación Id	Descripción			
EVF-0007	Transferencia a un dispositivo externo tres ejecuciones distintas al mismo tiempo tras un apagado del dispositivo.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:32:C6	Aceleración	Correcto	Correcto	Correcto
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Ordenador portátil	Correcto	2009_04_17_14_21_32.txt		7 KB
Ordenador portátil	Correcto	2009_04_17_14_23_15.txt		6 KB
Ordenador portátil	Correcto	2009_04_17_14_24_11.txt		3 KB

Tabla 23: Test de funcionamiento transferencia tras apagado

Los resultados de la tabla anterior reflejan el correcto funcionamiento de todos los procesos de transferencia.

5.3.2 Pruebas de Evaluación de la Calidad

Las pruebas de evaluación de la calidad de la aplicación consisten en el estudio de los datos obtenidos de los sensores, almacenados en la PDA y posteriormente transferidos al dispositivo externo.

La razón de estas pruebas radica en descubrir, mediante un análisis de las cadenas transferidas, si los datos son enviados de manera correcta. Para lo cual se establecen los siguientes criterios:

- Los sensores de aceleración envían datos con el formato definido en el apartado 2.2.2 de este documento. En él se establecía el requisito de la existencia de tres valores numéricos cada uno para un eje (X,Y,Z). Por tanto, un ejemplo de cadena correcta sería: “X = 123 Y = 456 Z= 789”, mientras que una errónea sería: “X = 123 Y = 456 Z=”
- Los sensores RFID envían cadenas con la longitud de las etiquetas que leen, es decir, diez caracteres hexadecimales. De este modo, un ejemplo de cadena correcta sería: “1234567890”, mientras que uno erróneo sería cualquiera de otro tamaño.

Según este criterio, las cadenas de datos que no cumplan alguna de las dos condiciones anteriores serán clasificadas como datos erróneos. La causa principal de los

mismos se debe a que, como se explicará más adelante, los datos viajan divididos en varios envíos y algún trozo puede perderse.

Para obtener resultados numéricos y comparables en todos los casos, se van a analizar los 1000 primeros datos leídos en cada iteración. De ellos se obtendrá el número de datos incorrectos y finalmente se obtendrá el porcentaje.

- **Comprobación de datos con un sensor de aceleración:**

Para realizar este test se va a establecer una comunicación con un solo sensor de aceleración y la posterior transferencia de los datos para su análisis. Inicialmente, se van a establecer unos valores de frecuencias predeterminados (100Hz de lectura del programa y 100Hz de envío de datos de los sensores) que posteriormente irán variando para ver su influencia en los datos.

Evaluación Id	Descripción			
EVC-0001	Análisis de los datos obtenidos tras la transferencia de una ejecución con un sensor de aceleración.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:18:C6:03	Aceleración	Correcto	Correcto	Correcto
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Ordenador portátil	Correcto	2009_04_18_11_12_01.txt		54 KB
Configuración				
Frecuencia lectura de datos		Frecuencia Configuración Sensor		
100 Hz		100 Hz		
Datos				
Número Datos	Datos Incorrectos	Porcentaje de Datos Incorrectos		
1000	9	0,9 %		

Tabla 24: Test de calidad con un sensor de aceleración

Como se puede apreciar en la anterior tabla el resultado estima un porcentaje de error en los datos de 0.9%. Esto es considerado bastante reducido y, por tanto, razonable. Posteriormente, se han repetido las ejecuciones con distintas frecuencias configuradas de envío de datos por parte del sensor (la frecuencia máxima es 220 Hz) cuyos resultados se exponen en la siguiente figura.

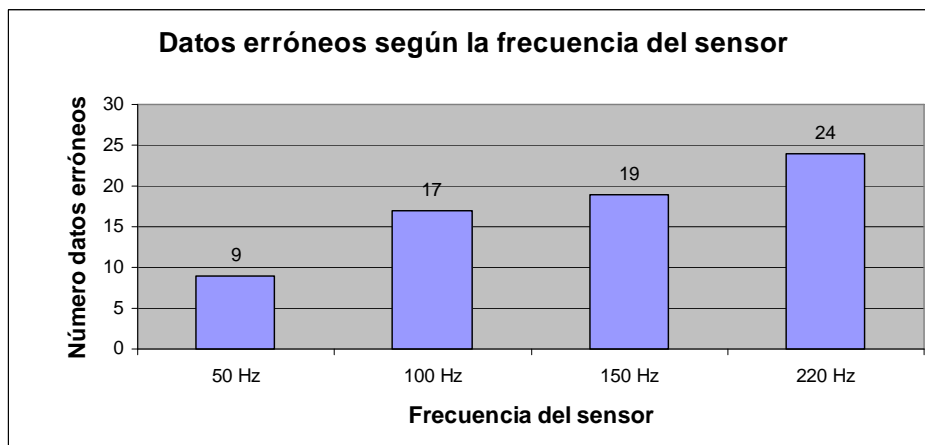


Figura 36: Número de datos erróneos según frecuencia del sensor

En ella se observa como el número de datos erróneos aumenta proporcionalmente con el incremento de la frecuencia de envío del sensor. Esto es debido a que el sensor envía datos a una velocidad superior a la que el dispositivo puede leer. Como los datos viajan fragmentados en varios envíos se producen cadenas incompletas o incorrectas.

De igual manera, se ha variado la frecuencia de lectura configurada en el programa para observar la variación consiguiente en el número de datos erróneos. Estableciendo de nuevo la frecuencia del sensor a 100 Hz, los resultados obtenidos en las ejecuciones se muestran en la siguiente figura.

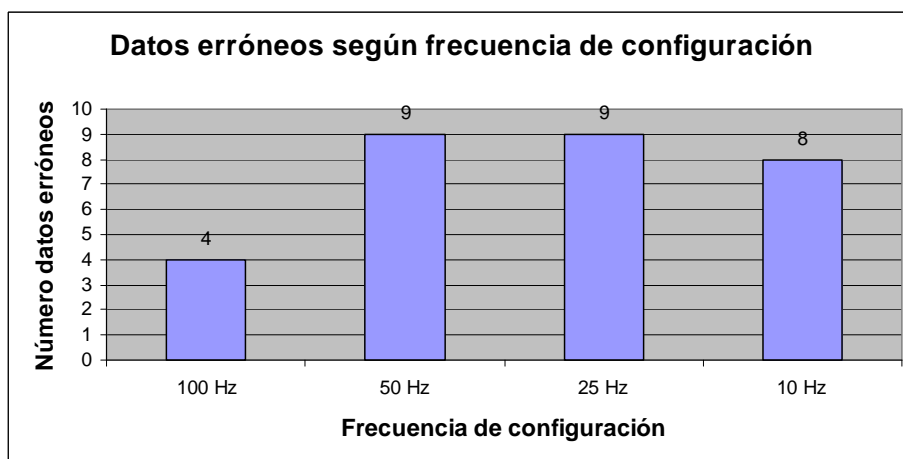


Figura 37: Número de datos erróneos según frecuencia de configuración

En el anterior gráfico se observa como la frecuencia de configuración que mejores resultados obtiene es 100 Hz. El principal motivo radica en que a mayor velocidad de lectura menos fragmentos de datos son descartados. Sin embargo, debido a

las limitaciones de procesamiento del dispositivo central de la aplicación, se ha comprobado que valores mayores de 100 Hz no son posibles.

- **Comprobación de datos con un sensor RFID:**

En esta prueba se comprueba la corrección de los datos obtenidos en las lecturas realizadas por un sensor RFID. Al contrario que en el test anterior, en este caso la frecuencia de configuración del sensor no es fácilmente configurable. Para ello, se necesitarían modificar características electrónicas del sensor para lo que no se tienen conocimientos. Por tanto, se ejecutarán todas las iteraciones sin cambiar dicho valor.

El resultado obtenido es óptimo, 0% de datos erróneos, y está reflejado en la siguiente tabla.

Evaluación Id	Descripción			
EVC-0002	Análisis de los datos obtenidos tras la transferencia de una ejecución con un sensor RFID.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:06:66:00:15:B8	RFID	Correcto	Correcto	Correcto
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Ordenador portátil	Correcto	2009_04_20_14_33_03.txt		34 KB
Configuración				
Frecuencia lectura de datos		Frecuencia Configuración Sensor		
100 Hz		-		
Datos				
Número Datos	Datos Incorrectos	Porcentaje de Datos Incorrectos		
1000	0	0 %		

Tabla 25: Test de calidad con un sensor RFID

- **Comprobación de datos con varios sensores:**

Para realizar esta prueba se utilizan los valores para las frecuencias que anteriormente han proporcionado mejores resultados, es decir, una frecuencia de envío de datos del sensor de 50 Hz y una frecuencia de lectura configurada en la aplicación de 100 Hz. Como se recoge en la siguiente tabla se conecta con tres sensores de aceleración y se comprueban los datos del fichero transferido.

Evaluación Id	Descripción			
EVC-0003	Análisis de los datos obtenidos tras la transferencia de una ejecución con varios sensores.			
Conexiones				
Dirección Sensor	Tipo Sensor	Enlace	Conexión	Lectura de datos
00:A0:96:1B:30:95	Aceleración	Correcto	Correcto	Correcto
00:A0:96:18:C6:03	Aceleración	Correcto	Correcto	Correcto
00:A0:96:1B:30:D0	Aceleración	Correcto	Correcto	Correcto
Transferencia Fichero				
Dispositivo	Transferencia	Nombre Fichero		Tamaño Fichero
Ordenador portátil	Correcto	2009_05_05_15_06_15.txt		105 KB
Configuración				
Frecuencia lectura de datos		Frecuencia Configuración Sensores		
100 Hz		100 Hz		
Datos				
Número Datos	Datos Incorrectos	Porcentaje de Datos Incorrectos		
1000	12	1,2 %		

Tabla 26: Test de calidad con varios sensores

Como se puede observar en la tabla anterior, el porcentaje de datos incorrectos es bastante bajo y por tanto se puede considerar que la calidad de los mismos es aceptable.

Además, para comprobar que se recibe información de todos los sensores de manera repartida, se ha calculado el porcentaje de datos recibidos de cada uno de ellos. En este caso concreto, sobre un total de 2066 datos, el resultado es el mostrado a continuación.

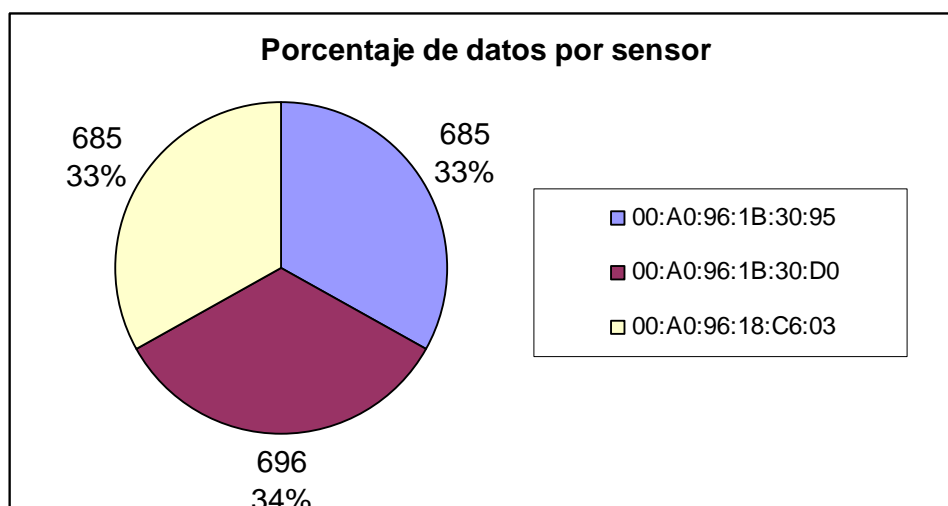


Figura 38: Porcentaje de datos por cada sensor

La proporción de datos de cada sensor es prácticamente la misma y, por tanto, se puede afirmar que las lecturas son equitativas.

5.3.3 Pruebas de Evaluación del Rendimiento

Las pruebas de evaluación del rendimiento de la aplicación consisten en la comprobación de la eficacia y la eficiencia de la misma. Para ello, se mide información relativa al tiempo. En concreto, se estudia la frecuencia de recogida de datos de los sensores que se produce en la PDA.

Con este objetivo, se realizarán ejecuciones variando tanto el número de dispositivos conectados como las frecuencias configuradas en los mismos y en el propio programa. Posteriormente, se analiza el número de datos leídos y se compara frente a la cantidad a priori estimada, obteniendo una valoración acerca de la frecuencia de lectura real. La duración de las pruebas se ha estipulado en 60 segundos, excepto las últimas de análisis de memoria y batería en las cuales el tiempo se alargará.

- **Estudio de la variación de la frecuencia de envío de datos de los sensores**

En este apartado se va a realizar un estudio de las diferentes frecuencias de configuración que se pueden introducir en los sensores para las ejecuciones con el programa. El motivo es establecer una serie de valores adecuados y compatibles con la aplicación. Dicha frecuencia puede ser configurada para valores de 0 a 220Hz en fracciones de 5Hz. Para concretar las pruebas se han escogido un subconjunto de estos valores con el fin de obtener un análisis completo. En concreto, se ejecutará con 220, 150, 100 y 50 Hz. No se ha considerado la posibilidad de que los distintos sensores que componen una ejecución tengan diferentes frecuencias. Esto es debido a que la

recepción de datos en la PDA podría ser heterogénea (recibiría más datos de unos sensores que de otros) e incluso a que alguno de ellos pudiera monopolizarla.

Las pruebas se realizarán con los sensores de aceleración, puesto que como se comentó anteriormente, la frecuencia de los sensores RFID no va a ser variada. Para ello, se ha escogido un sensor concreto y se han efectuado cuatro ejecuciones idénticas con la excepción del valor de la frecuencia de envío perteneciente a las cantidades anteriormente indicadas. Por otro lado, la frecuencia de lectura de datos del programa ha sido fijada como 100Hz para comparar adecuadamente los resultados obtenidos.

En los dos primeros tests (220 y 150 Hz) se han producido errores en la comunicación con los dispositivos. Las conexiones se han realizado correctamente y el inicio de la recepción de datos también. Sin embargo, tras el paso de un tiempo aproximado de 10 segundos, los sensores se desconectan por sí solos. Para comprobar que en este error no influye la aplicación, se han repetido ambas ejecuciones pero estableciendo la comunicación con el ordenador mediante un terminal, tal y como se explica en el Anexo B de este documento. En efecto, el mismo problema sucede. Por lo tanto, se obtiene la conclusión de que para estos valores de frecuencia el comportamiento de los sensores puede ser inesperado provocando un rendimiento inadecuado.

En cambio, en las otras dos pruebas (100 y 50 Hz), todo el proceso se ha realizado con normalidad y no se han producido errores. Debido a esta situación, para las futuras ejecuciones se escogerán valores para la frecuencia de envío de datos de los sensores de 50 o 100Hz.

- **Estudio del estado de los sensores**

En este apartado se ha realizado un análisis de las capacidades de los sensores sin utilizar la aplicación de este proyecto con el fin de comprobar las frecuencias de envío de los sensores y la futura influencia del programa en el rendimiento del sistema.

Para ello, se ha utilizado un ordenador portátil con conexión Bluetooth para establecer comunicación con los sensores y recoger la cantidad de datos leídos en un cierto tiempo, siguiendo el proceso detallado en el Anexo B de este documento. En concreto, se van a visualizar los datos recibidos de los sensores en 5 segundos para las frecuencias de envío de 50 y 100 Hz. Con el objetivo de evitar errores de precisión en el cálculo de dicho intervalo de tiempo, se repetirá cada ejecución en cuatro ocasiones y se hallará la media de los valores obtenidos. Los resultados se muestran en la siguiente tabla.

Sensor		Datos en ejecuciones				Media	Frecuencia real
Dirección	Frecuencia	1	2	3	4		
00:A0:96:1B:30:95	50 Hz	249	253	247	252	250,3	50,1 Hz
	100 Hz	502	499	500	495	499,0	99,8 Hz
00:A0:96:18:C6:03	50 Hz	255	238	248	246	246,8	49,4 Hz
	100 Hz	483	509	480	497	492,3	98,5 Hz
00:A0:96:1B:30:D0	50 Hz	231	217	234	240	230,5	46,1 Hz
	100 Hz	456	470	480	467	468,3	93,6 Hz
00:A0:96:17:A4:CD	50 Hz	226	241	237	236	237,5	47,5 Hz
	100 Hz	468	480	486	470	476	95,2 Hz
00:A0:96:1B:32:C6	50 Hz	252	255	243	250	250	50 Hz
	100 Hz	491	496	493	491	492,8	98,6 Hz

Tabla 27: Test de rendimiento de los sensores con el ordenador

En la anterior tabla se aprecian diferencias mínimas en el rendimiento de cada sensor. Además, se obtienen resultados que se acercan notablemente a los valores esperados en cada configuración. Si se comparan las ejecuciones de 100Hz de frecuencia de envío de datos se obtiene una gráfica como la siguiente.

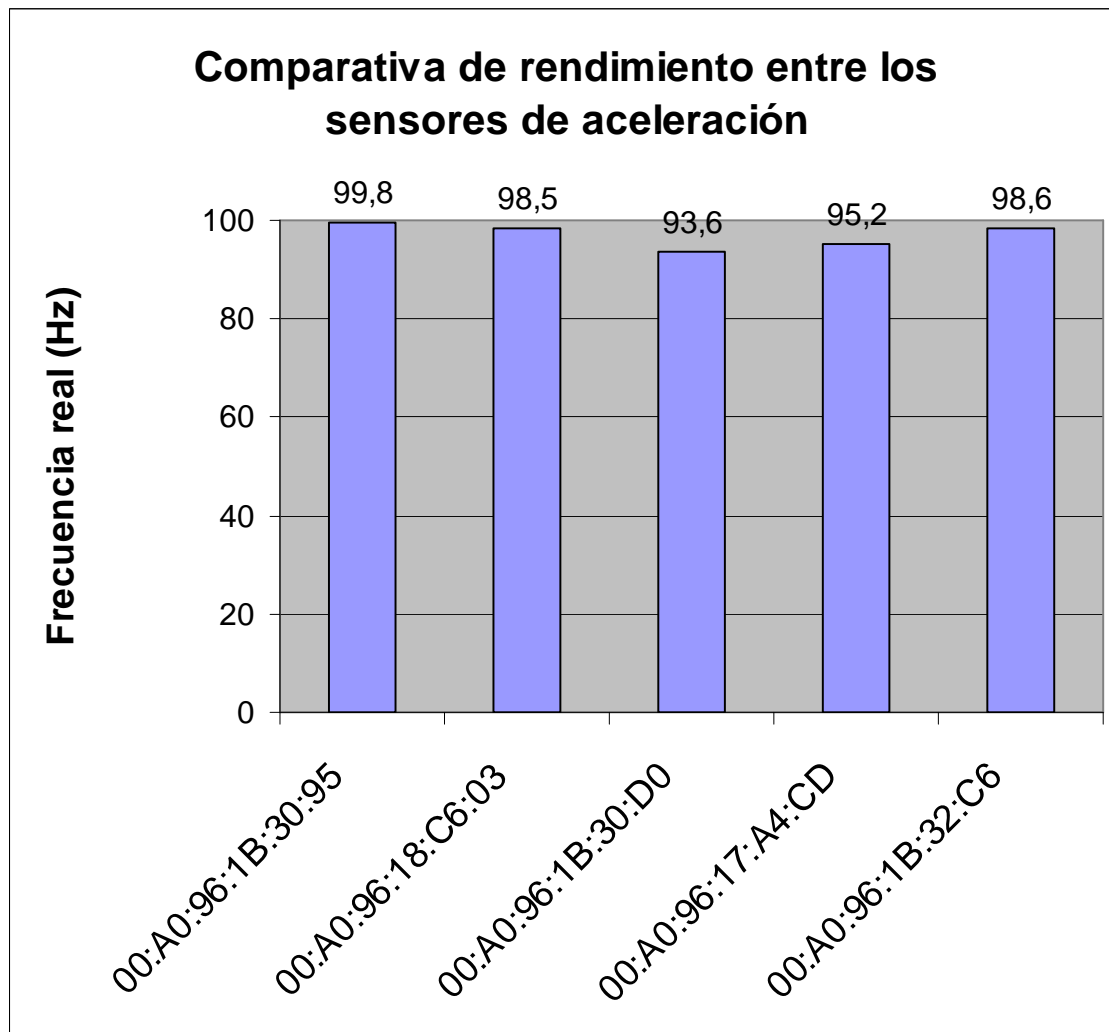


Figura 39: Comparativa del rendimiento de los sensores de aceleración con el PC

Por todo lo anterior, se puede afirmar que el estado de los sensores es adecuado y que funcionan correctamente.

- **Estudio de los sensores por separado con la aplicación**

A continuación se practica una comparación del rendimiento de los distintos sensores de aceleración disponibles. Para ello, se realiza una misma ejecución con cada uno de ellos y se analizan los resultados obtenidos. Tanto la frecuencia de envío de datos configurada en los sensores como la de lectura del programa se fijan en 100Hz en todos los casos. Los resultados obtenidos se recogen en las siguientes tablas.

Evaluación Id		Descripción				
EVR-0001		Comprobación del rendimiento del sensor 00:A0:96:1B:30:95				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	1497	24,9	24,9 Hz	2009_06_18_16_55_13.txt

Tabla 28: Test de rendimiento con conexión al sensor 00:A0:96:1B:30:95

Evaluación Id		Descripción				
EVR-0002		Comprobación del rendimiento del sensor 00:A0:96:18:C6:03				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	902	15,0	15,0 Hz	2009_06_18_16_56_44.txt

Tabla 29: Test de rendimiento con conexión a un sensor 00:A0:96:18:C6:03

Evaluación Id		Descripción				
EVR-0003		Comprobación del rendimiento del sensor 00:A0:96:1B:30:D0				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	1472	24,5	24,5 Hz	2009_06_18_17_17_47.txt

Tabla 30: Test de rendimiento con conexión a un sensor 00:A0:96:1B:30:D0

Evaluación Id		Descripción				
EVR-0004		Comprobación del rendimiento del sensor 00:A0:96:17:A4:CD				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	810	13,5	13,5 Hz	2009_06_18_17_04_36.txt

Tabla 31: Test de rendimiento con conexión a un sensor 00:A0:96:17:A4:CD

Evaluación Id		Descripción				
EVR-0005		Comprobación del rendimiento del sensor 00:A0:96:1B:32:C6				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	1479	24,6	24,6 Hz	2009_06_18_17_02_22.txt

Tabla 32: Test de rendimiento con conexión a un sensor 00:A0:96:1B:32:C6

Como podemos apreciar aparecen diferenciados dos grupos de valores. Por un lado, tres sensores muestran valores entorno a 25 Hz mientras que los otros dos se quedan solamente en unos 15. Esto se observa de manera gráfica en la siguiente figura.

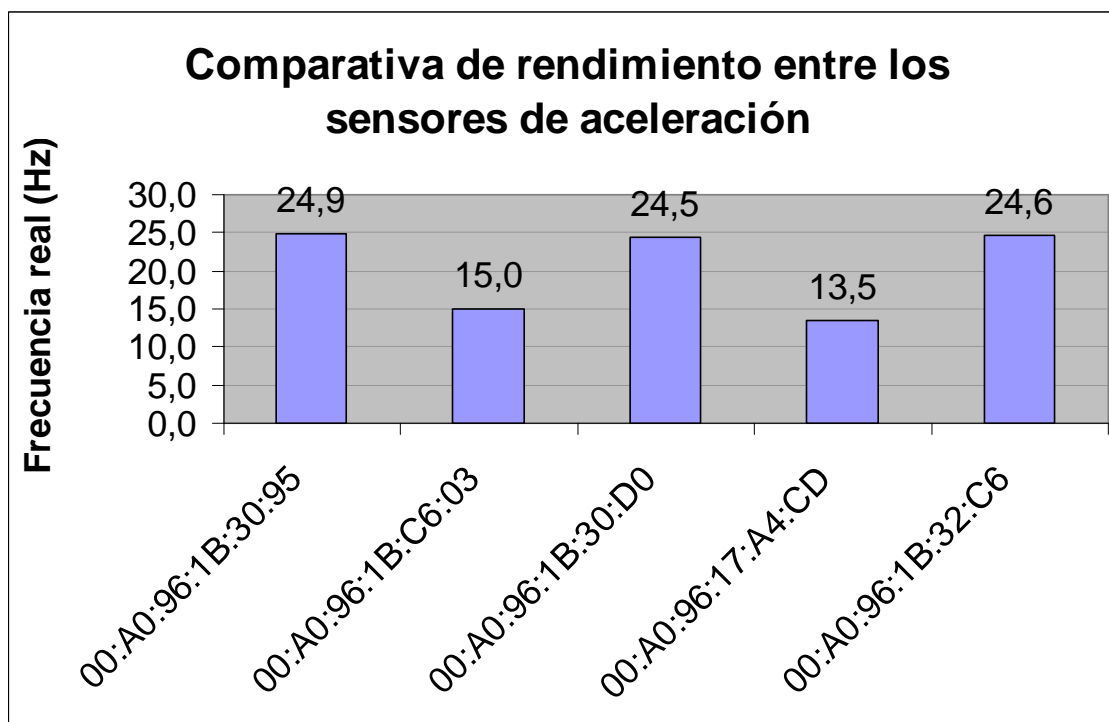


Figura 40: Comparativa del rendimiento de los sensores de aceleración con la PDA

Puesto que la ejecución del programa ha sido idéntica en todos los casos, estas diferencias en el valor de la frecuencia real son debidas únicamente al rendimiento de cada sensor. Es decir, bien al correcto funcionamiento del dispositivo o bien al estado de sus componentes.

De igual manera, se necesita comprobar si los resultados obtenidos por un mismo sensor en varias ejecuciones difieren en gran medida. Es decir, si se realizan varias ejecuciones repetidas con un sensor el resultado obtenido en cada una de ellas

debería ser bastante similar. Por ello, se ha escogido un sensor de aceleración, concretamente el que mejor resultado obtuvo en la anterior prueba, y se han realizado cuatro ejecuciones idénticas obteniendo los siguientes resultados.

Evaluación Id		Descripción				
EVR-0006		Comprobación de la variación del rendimiento del sensor de aceleración 00:A0:96:1B:30:95				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	1497	24,9	24,9 Hz	2009_06_18_16_55_13.txt
100 Hz	100 Hz	6.000	1503	25,1	25,1 Hz	2009_06_18_17_07_20.txt
100 Hz	100 Hz	6.000	1464	24,4	24,4 Hz	2009_06_18_17_08_31.txt
100 Hz	100 Hz	6.000	1500	25,0	25,0 Hz	2009_06_18_17_09_46.txt
100 Hz	100 Hz	6.000	1527	25,4	25,4 Hz	2009_06_18_17_11_17.txt

Tabla 33: Test de la variación del rendimiento del sensor 00:A0:96:1B:30:95

De nuevo, se muestra gráficamente la información principal recogida en la anterior tabla en la siguiente figura.

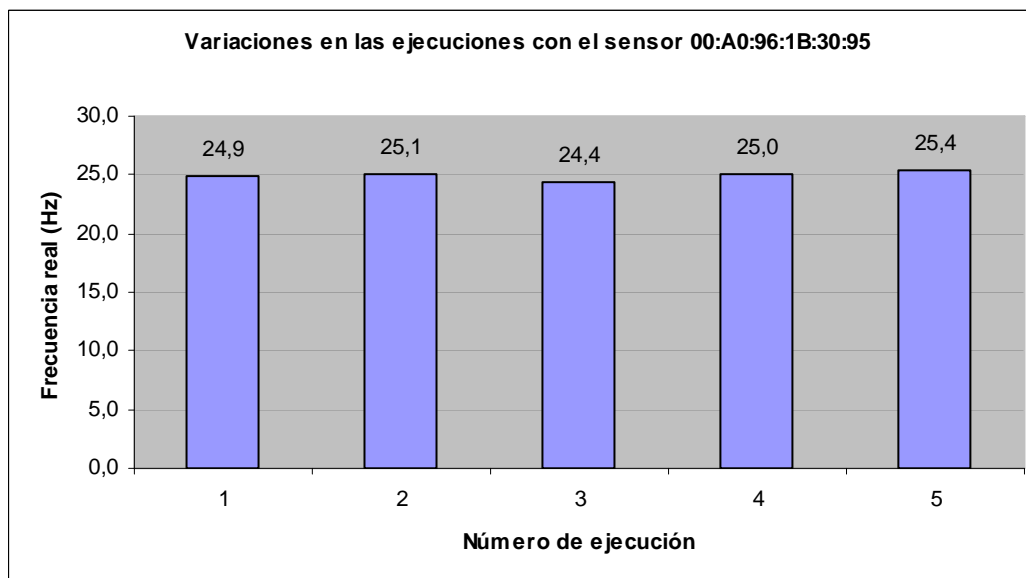


Figura 41: Variaciones en distintas ejecuciones con un sensor de aceleración

En ella se muestran una desviación mínima en los valores obtenidos. Por tanto, se demuestra que el comportamiento de los sensores no varía mucho en las distintas ejecuciones que se realicen con ellos.

- **Conexión con un sensor de aceleración**

El siguiente test permite analizar los resultados obtenidos conectando con un sensor en relación con la variación de las frecuencias de envío y lectura de datos que pueden ser configuradas. Para ello, se ejecuta en cuatro ocasiones, cada una de ellas con una combinación de valores distinta. La siguiente tabla muestra la información obtenida.

Evaluación Id		Descripción				
EVR-0007		Se realizan varias ejecuciones con conexión a un sensor de aceleración (00:A0:96:1B:30:D0) para comprobar la frecuencia de recogida de datos y su variación en relación con la configuración del sensor.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	6.000	1.452	24,2	24,2 Hz	2009_06_13_13_44_22.txt
50 Hz	100 Hz	6.000	1.395	23,2	23,2 Hz	2009_06_13_14_01_19.txt
100 Hz	50 Hz	3.000	1.299	43,3	21,6 Hz	2009_06_13_13_48_56.txt
50 Hz	50 Hz	3.000	1.272	42,4	21,2 Hz	2009_06_13_13_55_32.txt

Tabla 34: Test de rendimiento con conexión a un sensor de aceleración

Como se puede apreciar en la tabla anterior, el número real de datos obtenidos se aleja notablemente de los esperados, sobre todo en los ejemplos de frecuencia de programa de 100Hz. En concreto, la frecuencia máxima obtenida es de aproximadamente 24 lecturas por segundo. Esto se debe a las restricciones de procesamiento del dispositivo central de la aplicación ya que no puede ejecutar con más velocidad y multitud de datos no pueden ser tratados.

Además, se desprende una variación mínima en la frecuencia real de lectura de los datos con respecto de la frecuencia de los sensores, mientras que existe una disminución notable entorno a 2Hz en relación con la frecuencia de configuración del programa.

- **Conexión con dos sensores de aceleración**

En este caso se van a realizar las mediciones estableciendo conexión simultánea con los sensores cuyas direcciones Bluetooth son 00:A0:96:1B:30:D0 y 00:A0:96:1B:32:C6. Los datos de las ejecuciones se muestran en la siguiente tabla.

Evaluación Id		Descripción				
EVR-0008		Se realizan varias ejecuciones con conexión con dos sensores de aceleración (00:A0:96:1B:30:D0, 00:A0:96:1B:32:C6) para comprobar la frecuencia real de recogida de datos.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	12.000	2.337	19,5	19,5 Hz	2009_06_17_21_23_31.txt
50 Hz	100 Hz	12.000	2.259	18,8	18,8 Hz	2009_06_17_21_28_22.txt
100 Hz	50 Hz	6.000	2.022	33,7	16,8 Hz	2009_06_17_21_25_16.txt
50 Hz	50 Hz	6.000	2.142	35,7	17,8 Hz	2009_06_17_21_29_26.txt

Tabla 35: Test de rendimiento con conexión con dos sensores de aceleración

De nuevo, los valores de frecuencia real de datos obtenidos son bastante inferiores a los estimados y por tanto la cantidad de datos recogidos es menor de la esperada. Además, se puede mostrar cómo baja el rendimiento respecto a la anterior prueba puesto que con un sensor de aceleración por separado se obtienen 1.452 lecturas mientras que con dos juntos obtendríamos unas 1.164 de cada uno.

Por otro lado, se va a analizar la influencia en el rendimiento del hecho de que los datos que van siendo capturados de los sensores sean mostrados en la pantalla del dispositivo. Con este propósito se ha repetido la primera ejecución pero en esta ocasión se ha seleccionado la opción de mostrar los datos en el formulario de ejecución según van siendo leídos. Los datos se recogen en la siguiente tabla comparándolos con los de la anterior.

Evaluación Id		Descripción				
EVR-0009		Se realizan dos ejecuciones con conexión con dos sensores de aceleración (00:A0:96:1B:30:D0, 00:A0:96:1B:32:C6) para comprobar el retardo provocado por mostrar los datos por pantalla.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	12.000	2.337	19,5	19,5 Hz	2009_06_17_21_23_31.txt
100 Hz	100 Hz	12.000	2.247	18,7	18,7 Hz	2009_06_24_19_02_30.txt

Tabla 36: Test de rendimiento de comprobación del retardo por mostrar los datos

En efecto, el rendimiento es inferior, sin embargo, la diferencia no es tan elevada como podría esperarse debido a la gran cantidad de datos que debe mostrar y por tanto la numerosas veces que debe refrescar la pantalla. Por consiguiente, pese a no ser lo recomendado se considera una alternativa aceptable realizar las ejecuciones mostrando los datos por pantalla.

- **Conexión con tres sensores de aceleración**

En esta ocasión se añade un nuevo sensor, el dirección Bluetooth 00:A0:96:17:A4:CD, a las pruebas para comprobar los valores obtenidos estableciendo la conexión con tres sensores de aceleración. Los resultados se muestran a continuación.

Evaluación Id	Descripción					
EVR-0010	Se realizan ejecuciones con conexión con tres sensores de aceleración (00:A0:96:1B:30:D0, 00:A0:96:1B:32:C6, 00:A0:96:17:A4:CD) para comprobar la frecuencia real de recogida de datos.					
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	18.000	3.021	16,8	16,8 Hz	2009_06_17_22_07_49.txt
50 Hz	100 Hz	18.000	2.748	15,2	15,2 Hz	2009_06_17_21_58_35.txt
100 Hz	50 Hz	9.000	2.736	30,4	15,2 Hz	2009_06_17_22_11_00.txt
50 Hz	50 Hz	9.000	2.775	30,8	15,4 Hz	2009_06_17_22_01_52.txt

Tabla 37: Test de rendimiento con conexión con tres sensores de aceleración

Los valores de las frecuencias vuelven a estar por debajo de lo estimado y el rendimiento cada vez es menor.

- **Conexión con cuatro sensores de aceleración**

Para la siguiente prueba se ha utilizado, además de los anteriores, un nuevo sensor de aceleración, cuya dirección es 00:A0:96:18:C6:03, mediante el cual se han podido establecer cuatro conexiones simultáneas. Los resultados se recogen en la siguiente tabla.

Evaluación Id		Descripción				
EVR-0011		Se realizan ejecuciones con conexión con cuatro sensores de aceleración (00:A0:96:1B:30:D0, 00:A0:96:1B:32:C6, 00:A0:96:17:A4:CD, 00:A0:96:18:C6:03) para comprobar la frecuencia real de recogida de datos.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	24.000	3.390	14,1	14,1 Hz	2009_06_17_22_13_20.txt
50 Hz	100 Hz	24.000	3.387	14,1	14,1 Hz	2009_06_17_23_13_07.txt
100 Hz	50 Hz	12.000	3.174	26,4	13,2 Hz	2009_06_17_22_15_42.txt
50 Hz	50 Hz	12.000	3.381	28,1	14,1 Hz	2009_06_17_23_21_34.txt

Tabla 38: Test de rendimiento con conexión con cuatro sensores de aceleración

Los resultados continúan con la tendencia a la baja de los valores establecidos en un principio.

- **Conexión con cinco sensores de aceleración**

En este nuevo test se incluye un quinto sensor de aceleración que tiene la dirección Bluetooth 00:A0:96:1B:30:95. Los resultados se muestran a continuación.

Evaluación Id		Descripción				
EVR-0012		Se realizan ejecuciones con conexión con cinco sensores de aceleración (00:A0:96:1B:30:95, 00:A0:96:18:C6:03, 00:A0:96:1B:30:D0, 00:A0:96:17:A4:CD, 00:A0:96:1B:32:C6) para comprobar la frecuencia real de recogida de datos.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	30.000	3.810	12,7	12,7 Hz	2009_06_18_17_38_13.txt
50 Hz	100 Hz	30.000	3.424	11,4	11,4 Hz	2009_06_18_17_59_48.txt
100 Hz	50 Hz	15.000	3.615	24,1	12,1 Hz	2009_06_18_17_40_41.txt
50 Hz	50 Hz	15.000	3.453	23,0	11,5 Hz	2009_06_18_18_04_23.txt

Tabla 39: Test de rendimiento con conexión con cinco sensores de aceleración

Nuevamente se observan valores similares a las anteriores pruebas.

- **Conexión con un sensor RFID**

En esta prueba se va a establecer conexión con un sensor RFID con el fin de comprobar el número de lecturas que realiza este sensor. Para ello, se establece la comunicación y se sitúa una etiqueta RFID de lectura frente a la antena del sensor durante aproximadamente 60 segundos. Por el diseño de la aplicación, la frecuencia de configuración del programa es indiferente para las lecturas de este sensor. El resultado es mostrado a continuación.

Evaluación Id		Descripción				
EVR-0013		Se realiza una ejecución con conexión con un sensor RFID (00:06:66:00:15:B8) para comprobar su frecuencia de recogida de datos.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
---	100 Hz	6.000	897	14,9	14,9 Hz	2009_06_17_21_09_15.txt
---	50 Hz	3.000	916	30,5	15,2 Hz	2009_06_17_21_08_09.txt

Tabla 40: Test de rendimiento con conexión con un sensor RFID

El número de datos estimados se establece para igualar con los ejemplos en los que se usan sensores de aceleración debido a que se desconoce la frecuencia de lectura del sensor RFID. Por la función que realiza este sensor se considera la frecuencia real de datos como un valor adecuado puesto que simplemente debe comunicar si un objeto está situado a su alcance o no.

- **Conexión con un sensor RFID y uno de aceleración**

El objetivo de esta prueba consiste en comprobar si existe una influencia en las lecturas de los sensores de aceleración y RFID cuando se combinan en una misma ejecución. Para ello se ha realizado una conexión con la con ambos a la vez y se ha comparado con las realizadas anteriormente por separado. Los resultados se muestran a continuación.

Evaluación Id		Descripción				
EVR-0014		Se realiza una ejecución con conexión con un sensor RFID (00:06:66:00:15:B8) y uno de aceleración (00:A0:96:1B:30:D0) para comprobar si la combinación afecta a sus frecuencias de lectura.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	12.000	2.160	18,0	18,0 Hz	2009_06_17_21_13_25.txt
50 Hz	100 Hz	12.000	2.073	17,3	17,3 Hz	2009_06_18_17_24_53.txt
100 Hz	50 Hz	6.000	1.929	32,2	16,1 Hz	2009_06_17_21_18_09.txt
50 Hz	50 Hz	6.000	1.890	31,5	15,8 Hz	2009_06_18_17_26_23.txt

Tabla 41: Test de rendimiento con conexión al sensor RFID y uno de aceleración

Para efectuar un análisis de este test se escoge el primer valor de la tabla anterior, es decir, 2.160 lecturas. Analizando el contenido del fichero descomponemos este dato en 903 lecturas realizadas por el sensor RFID y 1.257 por el de aceleración.

A continuación se realiza una comparación con el número de datos obtenidos en las ejecuciones con los sensores por separado. Estos fueron obtenidos en anteriores pruebas y son 897 para el sensor RFID y 1.452 para el acelerómetro, haciendo un total de 2.349, casi un 9% más. Por tanto, en esta prueba los datos obtenidos por el sensor RFID son prácticamente las mismas, mientras que las lecturas del sensor de aceleración han sido considerablemente ralentizadas, lo que indica la influencia a la baja de la combinación de ambos sensores.

- **Conexión con un sensor RFID y cinco de aceleración**

Esta prueba es la más compleja puesto que se emplean todos los dispositivos disponibles. Además, se combinan los dos tipos de sensores puesto que se utiliza un RFID y cinco de aceleración. Los resultados se muestran a continuación.

Evaluación Id		Descripción				
EVR-0015		Se realiza una ejecución con conexión con un sensor RFID (00:06:66:00:15:B8) y cinco de aceleración (00:A0:96:1B:30:95, 00:A0:96:18:C6:03, 00:A0:96:1B:30:D0, 00:A0:96:17:A4:CD, 00:A0:96:1B:32:C6) para comprobar si la combinación afecta a sus frecuencias de lectura.				
Datos						
Frec Sensores	Frec Programa	Datos Estimados	Datos Reales	%	Frec real de datos	Fichero
100 Hz	100 Hz	36.000	3.432	9,5	9,5 Hz	2009_06_18_17_32_24.txt
50 Hz	100 Hz	36.000	3.732	10,4	10,4 Hz	2009_06_18_18_07_33.txt
100 Hz	50 Hz	18.000	3.825	21,2	10,6 Hz	2009_06_18_17_35_47.txt
50 Hz	50 Hz	18.000	3.633	20,2	10,1 Hz	2009_06_18_18_11_26.txt

Tabla 42: Test de rendimiento con conexión al sensor RFID y cinco de aceleración

Como se vio en las pruebas anteriores la combinación de los sensores provoca un descenso del rendimiento de la aplicación. En este caso se corrobora esta afirmación puesto que en él se obtiene el peor resultado.

- **Análisis de la ocupación de la memoria**

En esta sección vamos a estudiar la ocupación de la memoria tanto del dispositivo móvil como del ordenador externo al que se realizan las transferencias. Para ello, se realizan varias ejecuciones del programa y se comprueba la relación existente entre el número de lecturas guardadas en la base de datos de la aplicación, la memoria de la PDA y el tamaño de los ficheros creados tras la transmisión.

A continuación, se recogen los datos obtenidos tras cinco ejecuciones seguidas.

Lecturas	Lecturas acumuladas	Ocupación memoria PDA	Ocupación acumulada	Tamaño fichero PC	Tamaño acumulado
800	800	134 KB	134 KB	42 KB	42 KB
1500	2300	203 KB	337 KB	78 KB	120 KB
1500	3800	224 KB	561 KB	73 KB	193 KB
1500	5300	230 KB	791 KB	75 KB	268 KB
6500	11800	897 KB	1688 KB	330 KB	598 KB

Tabla 43: Ejemplos con su ocupación de memoria

Analizando los datos de la tabla anterior obtenemos una estimación de que por cada 1000 lecturas que se realicen se necesitarán aproximadamente 150 KB de almacenamiento en la PDA y 50 KB en el dispositivo externo.

A partir de estos datos y mediante las frecuencias obtenidas en los anteriores tests de rendimiento se puede estimar la ocupación de la memoria en base a las características de cada ejecución, esto es, el número de sensores que se conectan, la frecuencia de los sensores y la frecuencia de configuración del sistema. Estos cálculos se recogen en la siguiente tabla.

Num WiTilt	Num RFID	Frecuencia sensores (Hz)	Frecuencia programa (Hz)	Datos/ minuto	Ocupación PDA/minuto (KB)	Ocupación PC/minuto (KB)
1	0	100	100	1.452	217,8	72,6
1	0	50	100	1.395	209,3	69,8
1	0	100	50	1.299	194,9	65,0
1	0	50	50	1.272	190,8	63,6
2	0	100	100	2.337	350,6	116,9
2	0	50	100	2.259	338,9	113,0
2	0	100	50	2.022	303,3	101,1
2	0	50	50	2.142	321,3	107,1
3	0	100	100	3.021	453,2	151,1
3	0	50	100	2.748	412,2	137,4
3	0	100	50	2.736	410,4	136,8
3	0	50	500	2.775	416,3	138,8
4	0	100	100	3.390	508,5	169,5
4	0	50	100	3.387	508,1	169,4
4	0	100	50	3.174	476,1	158,7
4	0	50	500	3.381	507,2	169,1
5	0	100	100	3.810	571,5	190,5
5	0	50	100	3.424	513,6	171,2
5	0	100	50	3.615	542,3	180,8
5	0	50	500	3453	518,0	172,7
0	1	-	100	897	134,6	44,9
0	1	-	50	916	137,4	45,8
1	1	100	100	2.160	324,5	108,2
1	1	50	100	2.073	311,0	103,7
1	1	100	50	1.929	289,4	96,5
1	1	50	50	1.829	274,4	91,5
5	1	100	100	3.432	514,8	171,6
5	1	50	100	3.732	559,8	186,6
5	1	100	50	3.825	573,8	191,3
5	1	50	50	3.633	545,0	181,7

Tabla 44: Estimación de la ocupación de memoria

A modo de ejemplo, si se ejecuta el programa conectando con un sensor de aceleración con un valor 100Hz de frecuencia tanto en el sensor como en el programa, en un minuto se utilizarán 217,8 KB de memoria de la PDA. Si la ejecución tiene una duración aproximada de 5 minutos, se puede estimar una ocupación de aproximadamente 1089 KB. Si a su término realizamos una transferencia de datos a un ordenador, se generará un fichero cuya capacidad será de unos 363 KB ($72,6 \text{ ocupaciónPC/minuto} * 5 \text{ minutos}$).

Por todo esto, es necesario tener un control de la memoria disponible en la PDA. A partir de este dato podemos establecer las ejecuciones que vamos a poder realizar y sobre todo su duración. Si la memoria de la PDA se llena por completo, aparte de no

guardarse los nuevos valores, pueden producirse retardos e incluso errores en el comportamiento de la aplicación.

- **Análisis de la duración de la batería**

Esta última prueba se realiza para estudiar si las restricciones de batería del dispositivo móvil o de la pila de los sensores provocan limitaciones en la duración de las ejecuciones. Para ello, se ha llevado a cabo unas ejecuciones de un tiempo más elevado que las anteriores, con conexión a dos y tres sensores. Posteriormente, se han analizado los resultados obtenidos y recogidos en la siguiente tabla.

Duración	Número de Conexiones	Lecturas	Duración Transferencia	Fichero	Tamaño fichero
6 minutos	2 sensores aceleración	12206	30 segundos	2009_05_07_19_03_51.txt	619 KB
10 minutos	2 sensores aceleración	14667	50 segundos	2009_05_07_19_24_19.txt	744 KB
10 minutos	2 sensores aceleración y uno RFID	14579	45 segundos	2009_05_07_19_47_57.txt	695 KB

Tabla 45: Análisis de la duración de las baterías

En ella se aprecia el incremento de la duración del proceso de transferencia de datos a un dispositivo externo según aumenta el número de lecturas almacenadas en la base de datos.

En las pruebas realizadas no se han tenido problemas de descarga ni de la batería del dispositivo ni de las pilas de los sensores. Por tanto, se considera que en condiciones normales de carga se pueden realizar varias sesiones sin encontrar problemas de energía.

5.4 Resultados de la Evaluación

La aplicación final ha sido evaluada para comprobar la correcta ejecución de diferentes escenarios de ejecución. En primer lugar, es reseñable que la funcionalidad demostrada a lo largo de todas las pruebas es la deseada puesto que todos los subprocesos en los que se divide la aplicación han sido realizados correctamente.

En cuanto a la calidad de los datos obtenidos, se ha comprobado que el porcentaje de errores es lo suficientemente bajo como para no alterar los análisis que se

realicen a partir de los mismos. Además, se ha mostrado como los datos se recogen equitativamente de todos los sensores con los que se conecta.

Por otro lado, los test más profundos son los de rendimiento. En ellos es donde se han encontrado las principales limitaciones de la aplicación. No se han alcanzado los requisitos de rendimiento estipulados en un principio, es decir, las máximas frecuencias de lectura alcanzadas se hallan lejos de las deseadas. Esto es debido principalmente a las características del dispositivo móvil, el cual, no posee la suficiente capacidad de procesamiento para lograr los objetivos.

Se han realizado ejecuciones suficientes para comprobar que la frecuencia de lectura varía en función del número de sensores conectados. En la siguiente figura se puede apreciar cómo evolucionan estos valores.

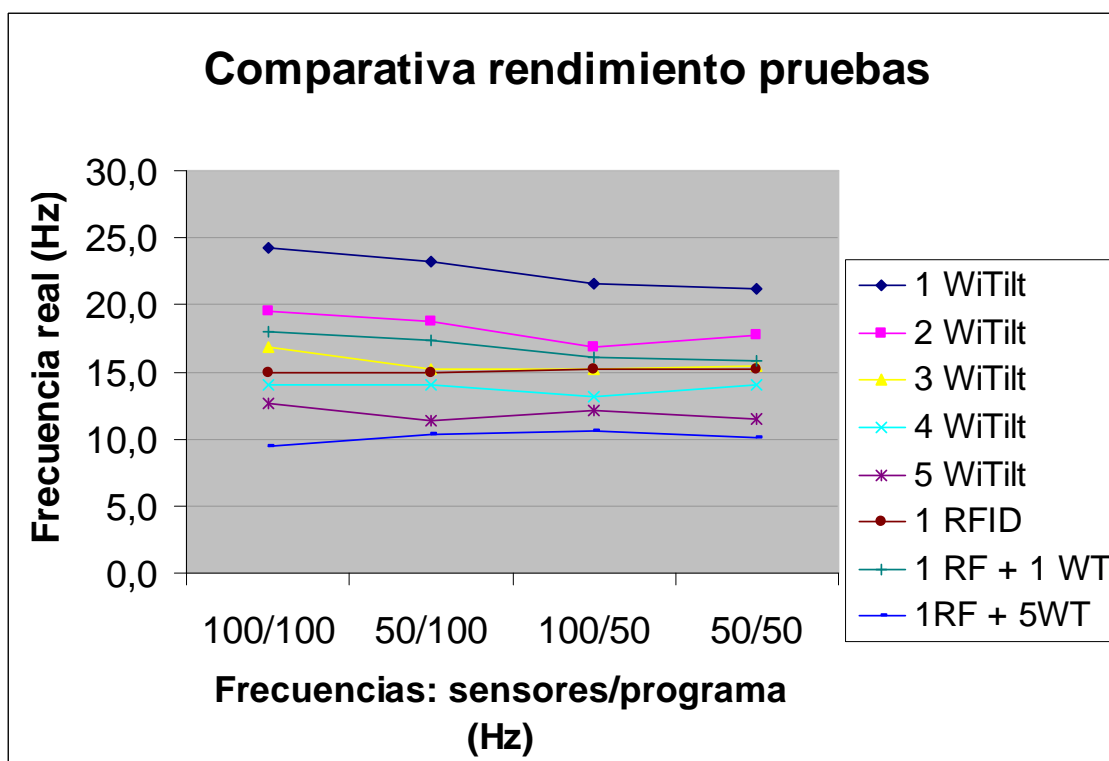


Figura 42: Comparativa frecuencias según número de sensores

Lo primero que debe recalcar es que han podido ser realizadas pruebas utilizando simultáneamente los seis sensores disponibles y no se han encontrado problemas aparentes de funcionamiento. Por otro lado, en la figura se aprecia un claro descenso del rendimiento de la aplicación según se aumenta el número de conexiones realizadas.

Para realizar un estudio más detallado del ejemplo más complejo que es el que establece conexión con todos los sensores disponibles, se ha analizado y desglosado el fichero resultante de la segunda ejecución (2009_06_18_18_07_33.txt) para hallar la frecuencia particular de cada uno. Estos valores son los indicados abajo.

Sensor	Lecturas	Porcentaje
00:A0:96:1B:30:95	585	15,7
00:A0:96:1B:32:C6	620	16,6
00:A0:96:18:C6:03	534	14,3
00:A0:96:17:A4:CD	603	16,2
00:A0:96:1B:30:D0	606	16,2
00:06:66:00:15:B8	774	20,7

Tabla 46: Desglose de lecturas del test de rendimiento con todos los sensores

Se puede observar el reparto de las lecturas que se ha producido. Sin embargo, se muestra con mayor claridad en el siguiente gráfico.

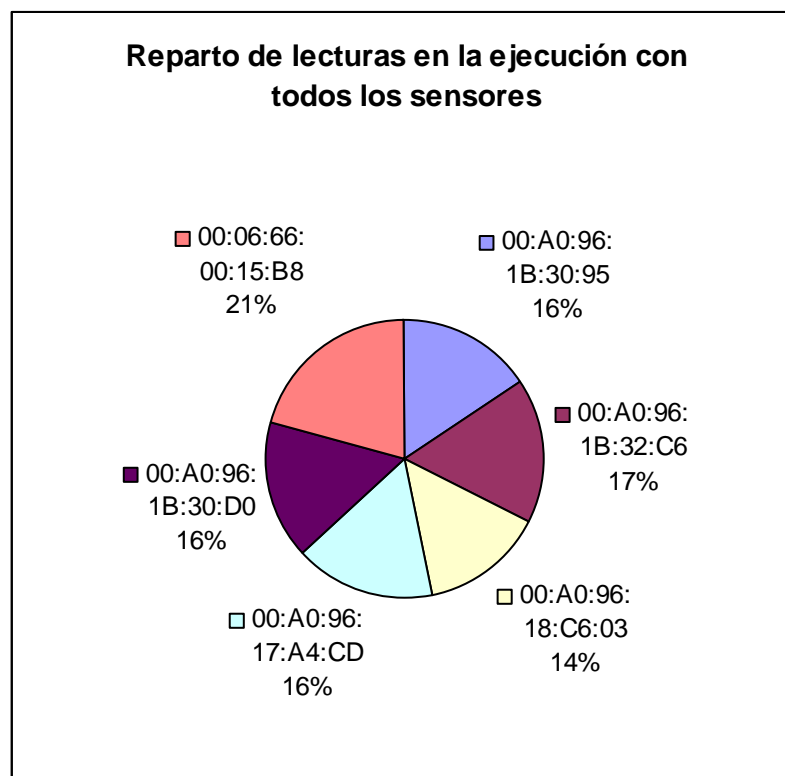


Figura 43: Comparativa frecuencias según número de sensores

Por un lado, se han obtenido un destacado número de lecturas del sensor RFID, esto se debe a que sus envíos no se dividen en trozos que el sistema debe ensamblar sino que manda las cadenas completas. Por otro lado, los sensores de aceleración proporcionan una cantidad de datos bastante similar, mostrando nuevamente un

rendimiento menor para el sensor 00:A0:96:18:C6:03. Esto indica que pese a que el número de datos obtenidos es inferior de lo esperado el reparto es equitativo.

Finalmente, se han analizado positivamente aspectos de memoria del dispositivo y autonomía de la aplicación.

6. Conclusiones

En este apartado se recogen las principales conclusiones obtenidas tras la realización de este proyecto fin de carrera.

En general, el proceso de desarrollo de este trabajo se ha apoyado sobre los conocimientos adquiridos durante los estudios universitarios. Este proyecto ha supuesto un gran esfuerzo debido principalmente a dos motivos. El primero radica en la, al menos aparente, exclusividad del sistema propuesto, puesto que en la fase de investigación no se ha descubierto ninguna aplicación similar. Del mismo modo, a pesar de que la información encontrada ha sido elevada, es considerada insuficiente porque no se ajustaba al enfoque práctico requerido para ayudar al desarrollo del programa. Apenas existe documentación acerca del uso de la tecnología Bluetooth en el dispositivo central de este sistema.

La propia naturaleza del sistema realizado, definida desde su planteamiento, lo sitúa dentro del desarrollo actual de las tecnologías. Se puede considerar como un proyecto innovador. Por un lado, emplea tanto dispositivos modernos como una de las tecnologías de comunicación más utilizadas actualmente. Por otro, se integra en la actual tendencia a la creación de sistemas inalámbricos.

El programa creado posee la particularidad de ser un sistema abierto a multitud de aplicaciones. Establece una comunicación entre dispositivos y obtiene el registro de los datos procedentes de ellos. A partir de esta situación, el escenario de ejecución de la aplicación y la posterior utilización que se proporcione a los datos capturados pueden ser de una amplia variedad y, por tanto, el sistema se puede adaptar a multitud de situaciones. Algunas de ellas son indicadas en el apartado de trabajos futuros de este documento.

Un aspecto a recalcar es la extensa cantidad de material empleado. Multitud de dispositivos se han utilizado con el fin de dominar el tema tratado. Varios sensores de aceleración, un sensor RFID, una PDA y un ordenador portátil componen el conjunto de dispositivos hardware del sistema final. Además, como está documentado en los anexos de este documento, una antena GPS, un mando de la videoconsola Wii y un teléfono móvil han sido empleados durante el proceso de estudio del planteamiento del problema. Con ello, se ha conseguido que la aplicación generada deje abierto un futuro desarrollo para la adaptación a nuevos dispositivos que reúnan las condiciones suficientes para ser introducidos en el sistema creado.

Para la creación de este proyecto, un proceso clave ha sido el de la formación. Ha sido necesario obtener un profundo conocimiento para explotar al máximo las

posibilidades tanto de los dispositivos como de las tecnologías empleadas. Por consiguiente, la investigación realizada ha sido larga y meticulosa con el fin de reunir la máxima información posible para evaluar las diferentes posibilidades y escoger la considerada como la que mejor se adapta a los requisitos del sistema.

El análisis del estado del arte se ha llevado a cabo de dos maneras. La primera de ellas consiste en una investigación general de cada uno de los aspectos a estudiar. La segunda y posterior, se ha basado en una búsqueda más particular centrada en los dispositivos concretos proporcionados para el sistema. Esta fase ha repercutido positivamente en el autor del proyecto aumentando el aprendizaje de nuevos conceptos y reforzando el conocimiento de otros previamente conocidos.

A la hora de diseñar la aplicación, se ha considerado la opción de evitar, en la medida de lo posible, la dependencia de la plataforma de implementación de la misma. Es decir, se han agrupado en módulos las operaciones similares, o que utilizan los mismos recursos específicos del dispositivo móvil, para su implementación por separado. De esta manera, se reduciría el impacto en la portabilidad del programa a otro terminal.

En la fase de diseño, se ha empleado UML puesto que es considerado un patrón universal de diseño. Además, ha servido para ilustrar el diseño de manera gráfica mediante los diagramas incluidos en este documento con el fin de facilitar la comprensión al lector.

La implementación ha sido orientada al dispositivo móvil proporcionado para la realización del proyecto, la PDA Tungsten E2 de Palm. Se ha utilizado una plataforma desarrollada por la empresa fabricante que ha facilitado la creación de la aplicación. Se ha codificado mediante lenguaje C, el cual es mundialmente conocido y proporciona las funcionalidades suficientes para la programación del dispositivo. Además, la limpieza y el comentado del código ha sido clave para la comprensión del mismo y reducir el tiempo dedicado a la depuración.

El sistema creado se puede considerar como un sistema bastante complejo debido a los numerosos factores que condicionan las ejecuciones. La utilización de abundante material implica la aparición de posibles problemas provocados por el correcto funcionamiento de cada dispositivo, el estado de la batería o pilas de los mismos o incluso simplemente de la disponibilidad de todos ellos. Esto último, ha sido un factor de retraso en la duración de las tareas tanto de implementación como de evaluación.

La fase de evaluación ha sido fundamental a la hora de comprobar la eficacia y eficiencia del sistema creado. Las pruebas de funcionalidad han corroborado el correcto funcionamiento de la aplicación ya que se han superado la totalidad de las mismas. La calidad de los datos obtenidos de los dispositivos se puede considerar adecuada teniendo en cuenta la alta velocidad de lectura y el posible ruido existente en las conexiones. El gran aspecto negativo de la aplicación es el rendimiento de la misma. Es considerado como aceptable, sin embargo, las frecuencias de lectura deseadas no han podido ser alcanzadas. Esto se debe principalmente a las limitaciones de los recursos de procesamiento del dispositivo móvil central del proyecto. Por otro lado, sería necesario exportar el programa a otra plataforma de implementación para comprobar cómo influye el código generado en el retardo de las lecturas.

El presupuesto del proyecto presenta un precio superior al estimado. Esto es debido a que, como se comentó anteriormente, la fase de investigación se ha extendido para considerar las diferentes posibilidades. Además, la implementación del programa se realizó incluyendo comprobaciones en todo momento de que el sistema tenía un funcionamiento correcto. Esta fase ha sido bastante costosa debido a no disponer de un simulador para las conexiones Bluetooth, sino que debía ser cargada y ejecutada en el dispositivo en cada ocasión.

Para concluir, se debe comentar que la constante evolución tecnológica que se está llevando a cabo en estos tiempos, provoca la aparición de nuevos dispositivos y productos que mejoran las condiciones y capacidades de los actuales. Esto puede derivar en una apariencia atrasada de un sistema creado. Sin embargo, combinando el planteamiento y diseño realizado para este sistema con la investigación en las futuras nuevas tecnologías se puede perfeccionar la aplicación para que se aproveche de los nuevos adelantos.

7. Trabajos futuros

En este capítulo se enumeran una serie de trabajos de ampliación de este proyecto fin de carrera que pueden ser realizados tras su conclusión.

En primer lugar, y como ha sido comentado en anteriores apartados de este documento, las limitaciones del dispositivo móvil provocan que el rendimiento de la aplicación sea mejorable. Una de las soluciones consiste en cambiar la plataforma de ejecución, es decir, utilizar otro dispositivo móvil más potente. Esto implicaría cambios en la fase de implementación de la aplicación, puesto que las librerías propias del fabricante para manejo de funcionalidades como las conexiones Bluetooth cambian entre las diferentes marcas y lenguajes de programación.

La aplicación no utiliza procesos o hilos para capturar los datos debido a que el dispositivo central no permite su uso. Cambiando de dispositivo central a otro que permita la programación multihilo se podría mejorar notablemente el rendimiento del programa.

Otro problema surgido por la utilización de dicho dispositivo central ha sido la imposibilidad de establecer conexión con una antena GPS. Tras intentar la comunicación con dos antenas distintas, una de ellas la proporcionada junto con la PDA por Palm y otra proveniente de otra compañía, utilizando varios métodos distintos, vía Bluetooth y vía puerto serie, no se encontró la manera de comunicarse entre la PDA y estos dispositivos GPS. Por esta razón, se desestimó incluir esta funcionalidad en el sistema. Por ello, una ampliación del programa consistiría en la inclusión de una antena GPS con Bluetooth en el sistema con el fin de obtener las coordenadas geográficas en todo momento y poder ubicar la ejecución de la aplicación y los movimientos de translación del usuario. Para ello, se debería poder conectar con el dispositivo de manera automática y realizar lecturas periódicas de manera similar a las lecturas de datos de los sensores.

Para mejorar el rendimiento y la calidad de las lecturas, en el caso de que el dispositivo móvil fuese capaz de capturar más rápidamente los datos, puede realizarse una versión del sistema con el tratamiento de los datos obtenidos de los sensores de aceleración en modo binario. Sería necesario estudiar cómo van codificados los datos en esta versión, el tamaño de los mismos y el tratamiento necesario para hacerlos legibles.

Con el objetivo de ampliar las posibilidades del sistema, una medida interesante consistiría en la inclusión de un mando de la videoconsola Wii como sensor de aceleración. La aplicación debería ser transparente a los dispositivos que se conectan y comportarse de igual forma. Para ello, sería necesario comprobar los datos ofrecidos por el acelerómetro del mando, así como la velocidad de transmisión del mismo. Esto

supondría una rebaja en el costo del sistema, puesto que el precio del mando es bastante inferior al de los sensores utilizados en este proyecto.

Para extender la memoria del dispositivo móvil, una opción recomendable sería el almacenamiento de los datos en tarjetas de memoria extraíbles. De este modo, se eliminaría el proceso de transferencia de datos del programa, puesto que se podrían transferir directamente a través de la tarjeta. Para conseguirlo, la aplicación debería ser capaz de escribir en las tarjetas extraíbles en vez de en la memoria interna del terminal.

Anexos

En este apartado se introduce una serie de información auxiliar que completa el presente documento y, por tanto, la memoria de este proyecto.

A Manual de Usuario

En este apartado se introduce la práctica de la aplicación y se pretende explicar cómo debe realizarse una ejecución correcta, es decir, los pasos a seguir y la manera de operar para conseguir los resultados deseados.

A.1 Instalación del Ejecutable

La instalación en el dispositivo puede realizarse de dos maneras, dependiendo si se dispone del fichero ejecutable o de los ficheros con el código fuente de la aplicación.

La primera de ellas es la recomendada por su rapidez y comodidad. Consiste en la instalación directa del fichero ejecutable de la aplicación mediante una sincronización entre el ordenador de mesa o portátil y la PDA. Para ello, se necesita tener instalado en el ordenador el programa PalmOne Quick Install. La sincronización puede realizarse por cable, a través del puerto serie de la PDA y un puerto USB del ordenador, o por Bluetooth, habiendo configurado previamente esta conexión entre ambos.

La otra manera de realizar la instalación consiste en realizar una compilación a partir del código fuente de la misma y el fichero de definición de la interfaz. Para ello, se debe tener instalado el programa de desarrollo de aplicaciones para Palm denominado Garnet OS IDE. En él, se debe generar un nuevo proyecto estándar y dentro de su paquete principal insertar los ficheros acorde con el árbol de directorios de la siguiente figura.

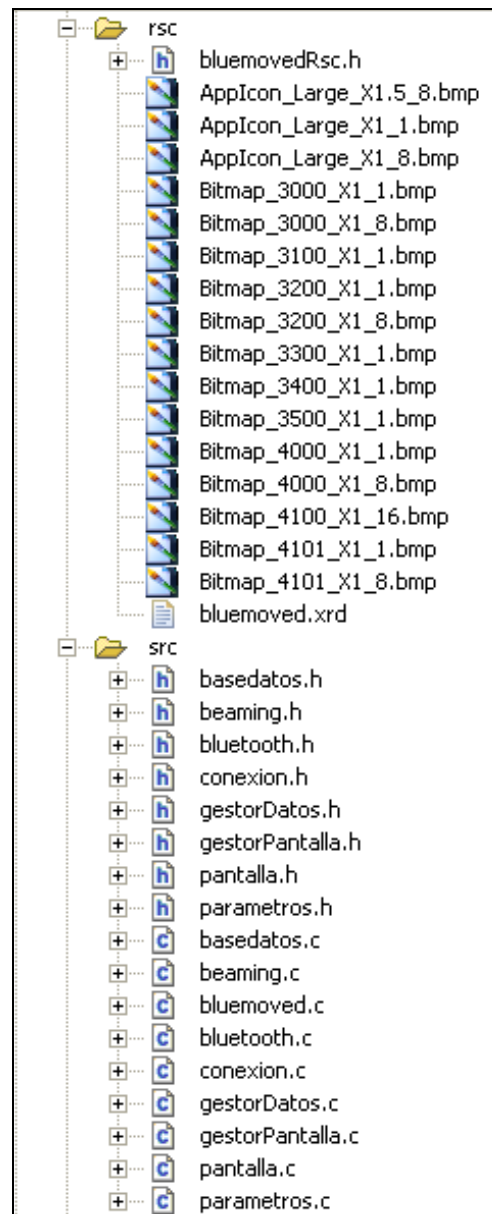


Figura 44: Árbol de directorios y ficheros

Los archivos con extensión “.bmp” son imágenes que se generan automáticamente al actualizar el fichero de definición de la interfaz, “bluemoved.xrd”.

Mediante una compilación del proyecto se genera el fichero ejecutable y se procede a la instalación siguiendo los pasos expuestos anteriormente.

A.2 Ejecución de la Aplicación

En este apartado se realiza un seguimiento de la aplicación acompañada de imágenes reales capturadas en una ejecución concreta de la misma.

El fichero ejecutable está configurado para que al ser instalado en un dispositivo, se aloje en una carpeta llamada PFC. Seleccionando esa carpeta en el menú principal y tocando la pantalla sobre el icono de la aplicación, ésta se abre y comienza la ejecución del programa con la pantalla de bienvenida.



Figura 45: Pantalla de Bienvenida

El menú de la aplicación, mostrado en la siguiente figura, es el que ofrece las diferentes acciones que puede realizar el usuario. Para seleccionar una de ellas basta con tocar en la pantalla sobre el texto que la compone.



Figura 46: Pantalla Menú

Para realizar una nueva lectura de datos la opción que debe ser escogida en el menú es “1. Nueva ejecución”. Inmediatamente, se muestra la siguiente figura que es la pantalla de configuración.

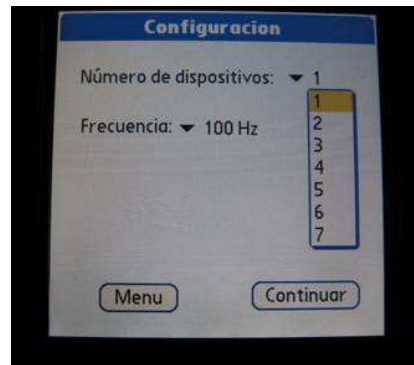


Figura 47: Pantalla de Configuración

En ella, el usuario elige el número de dispositivos y la frecuencia de lectura tocando directamente en la pantalla los valores de sus correspondientes listas desplegables. Tras seleccionar dichos valores y pulsando el botón “Continuar” se conduce al formulario de conexión mostrado a continuación.

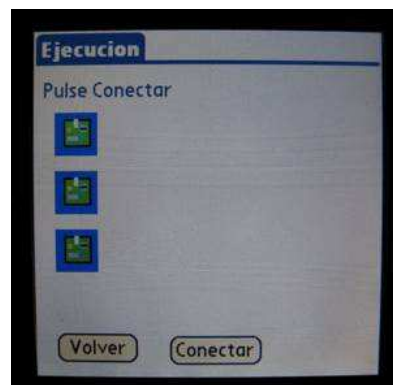


Figura 48: Pantalla de Conexión

Tras tocar el botón “Conectar”, se muestra un listado de las direcciones Bluetooth de los dispositivos que la máquina encuentra dentro de su alcance de radio. Además, se permite al usuario que los seleccione a través del cuadrado situado a la izquierda de cada uno. Si algún dispositivo no aparece en la lista, el usuario debe comprobar el correcto estado del mismo y pulsar el botón “Buscar más”.



Figura 49: Pantalla de Listado de Dispositivos

Cuando se tengan seleccionados los terminales deseados y mediante el botón “Ok” comienzan a establecerse las conexiones. Si los enlaces son creados correctamente se permite al usuario comenzar la lectura de datos a través del botón “Leer”.



Figura 50: Pantalla de Conexiones Establecidas

Se ofrece la posibilidad de visualizar en tiempo real los datos que van siendo capturados de los sensores a través del seleccionable “Mostrar valores” como se muestra a en la siguiente imagen.

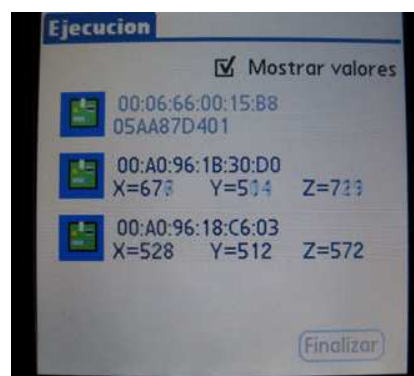


Figura 51: Pantalla de Muestra de Datos

Sin embargo, dicha visualización provoca un retardo en el rendimiento de la aplicación y debe ser utilizada únicamente como método de comprobación del establecimiento de las conexiones. Cuando el usuario decida finalizar la captura debe seleccionar el botón “Finalizar” e inmediatamente se mostrarán los datos que resumen el resultado de la operación.

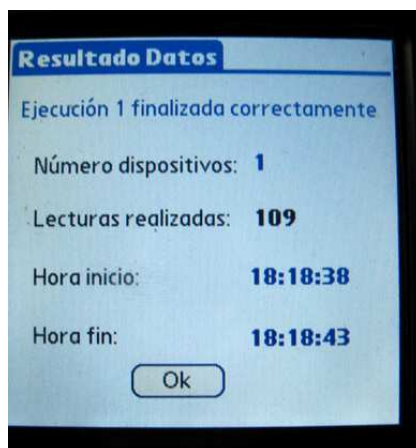


Figura 52: Pantalla de Resultados

A la hora de llevar a cabo una transferencia de datos a otro dispositivo externo, se debe seleccionar en el menú la opción número 2 etiquetada como “Transferir datos”. Se mostrará la siguiente pantalla para configurar el método de transmisión en la que el usuario deberá seleccionar la opción Bluetooth.



Figura 53: Pantalla de Transferencia por Bluetooth

Debe asegurarse que el dispositivo receptor está preparado para establecer la conexión Bluetooth. Para cada ejecución alojada en la base de datos se procede de la siguiente manera. Primero se muestra un listado, como el de la siguiente figura, de los

nombres de los aparatos Bluetooth que encuentra dentro de su alcance permitiendo al usuario escoger uno.



Figura 54: Pantalla de Listado de Dispositivos para la Transferencia

Una vez que el usuario escoge uno de la lista y selecciona el botón “Ok”, se muestra el número de ejecución actual y comienza el envío. El proceso efectúa el envío de todas las ejecuciones realizadas desde el último borrado de los datos. El tiempo de duración de la operación depende de la cantidad de datos a transferir.



Figura 55: Pantalla de Transferencia

Finalmente, se crea un fichero por ejecución en el dispositivo receptor y se cierra la comunicación. Una vez concluidas todas las transferencias, se permite al usuario volver al menú a través del botón “Finalizar” de la siguiente pantalla.



Figura 56: Pantalla de Finalización de Transferencia

Para realizar una limpieza de la base de datos, el usuario debe seleccionar en el menú la opción número 3 etiquetada como “Borrar datos” y, a continuación, se muestra el siguiente formulario de borrado.



Figura 57: Pantalla de Confirmación de Borrado de Datos

Seleccionando el botón “Si” ante la pregunta “¿Seguro que desea borrar los datos?” comienza el borrado. La duración de este proceso depende de la cantidad de datos almacenados en la base de datos. A su término, se permite al usuario volver al menú a través del botón “Ok”.



Figura 58: Pantalla de Finalización de Borrado de Datos

En adición, es posible recibir información acerca de las posibles acciones a realizar mediante la consulta del manual de usuario que se ofrece en la propia aplicación. Para ello el usuario debe escoger la opción 4 del menú “Manual de usuario”. A través de unos botones, se permite la navegación por los distintos formularios que lo componen. Incluye detalles para la correcta ejecución de cada una de las fases principales del programa. En la siguiente imagen se pueden observar dos pantallas de este manual, la primera muestra el menú de la información que contiene y la segunda un ejemplo de la información que se proporciona.

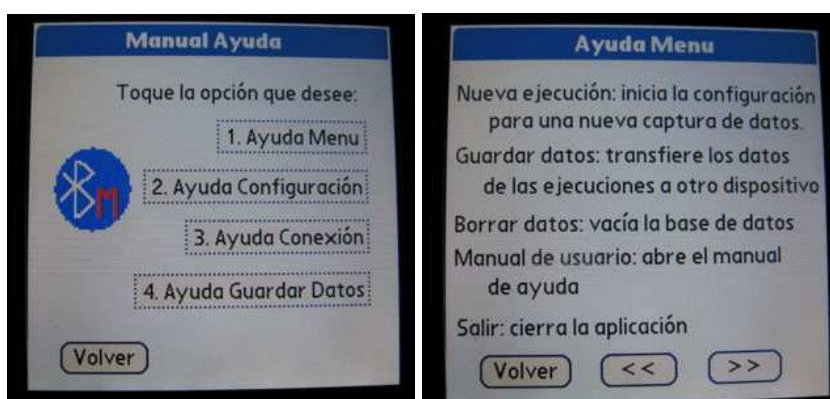


Figura 59: Pantallas de Ayuda

Por último, cuando el usuario selecciona la opción 5 del menú principal, “Salir”, la aplicación finaliza y se muestra la siguiente pantalla de despedida.



Figura 60: Pantalla de Despedida

Cuando el usuario pulsa el botón “Salir” del formulario anterior el programa se cierra.

A.3 Consejos para una Correcta Ejecución

A la hora de establecer las conexiones, si un dispositivo no aparece en la lista de aparatos Bluetooth al alcance de la PDA, es necesario comprobar si dicho dispositivo está encendido y si está correctamente configurado para ser utilizado. Si tras una ejecución correcta en la siguiente no aparece, puede deberse al estado de las baterías. Para comprobarlo, se recomienda apagar y volver a encender el dispositivo y realizar de nuevo la búsqueda. Si el estado de las pilas o batería es suficientemente bajo, es necesario la carga o el reemplazo de las mismas.

Tras haber establecido las conexiones y en el momento de comenzar la lectura de datos de los dispositivos, se recomienda la pulsación del botón Reset de los dispositivos. Se trata así, de evitar problemas provocados porque, en ocasiones y tras varias ejecuciones, puedan quedarse bloqueados y no se producirían los envíos.

Se recomienda el borrado de datos de ejecuciones que no se han realizado correctamente para que la base de datos del dispositivo no se llene provocando así restricciones o problemas posteriores.

Es recomendable a su vez la transferencia de los datos de las ejecuciones correctas. La memoria del dispositivo móvil es limitada y puede alcanzar su tope fácilmente con los consiguientes problemas que supondría.

Por último, relacionado con el dispositivo móvil, es aconsejable que el estado de la carga de la batería sea elevado debido a que las operaciones realizadas que emplean comunicaciones Bluetooth tienen un alto consumo.

B Configuración de Dispositivos Bluetooth

Para la realización del proyecto, se han utilizado varios dispositivos Bluetooth. Debido a que la aplicación se ha realizado de manera flexible, es decir, que se permita ejecutar con cualquier sensor de los modelos empleados en el desarrollo del programa, es posible adquirir nuevos conectores para añadirlos al sistema o sustituir los anteriores.

Para ello, la primera tarea que se debe llevar a cabo es comprobar si el funcionamiento de los sensores es correcto. El método más sencillo se basa en la sincronización directa con un ordenador, de sobremesa o portátil, que tenga conexión Bluetooth, ya sea interna o a través de un dispositivo externo. En la computadora, mediante el uso de un terminal, se va a verificar si el dispositivo puede ser empleado en el sistema.

En este anexo se va a ilustrar, paso a paso, cómo se ha desarrollado esta tarea con los sensores empleados en este proyecto. Para ello, se ha utilizado un ordenador portátil con el sistema operativo Windows Vista y un dispositivo externo Bluetooth conectado por puerto USB que permite a la computadora establecer conexiones a través de esta tecnología. Además, se ha utilizado la aplicación software denominada Hyperterminal [35], que consiste en una consola que muestra el estado y la comunicación en las diferentes conexiones que se configuren.

Para comenzar la operación se enciende el sensor Bluetooth y se enchufa el dispositivo USB, previamente configurado siguiendo su correspondiente manual de instrucciones. El siguiente paso consiste en agregar un puerto COM al ordenador que identifique la conexión con el sensor. Para ello, se pulsa con el botón derecho sobre el icono de Bluetooth, situado en la barra de tareas de la pantalla, y posteriormente se selecciona la opción “Agregar un dispositivo Bluetooth”. La siguiente ventana es mostrada a continuación.

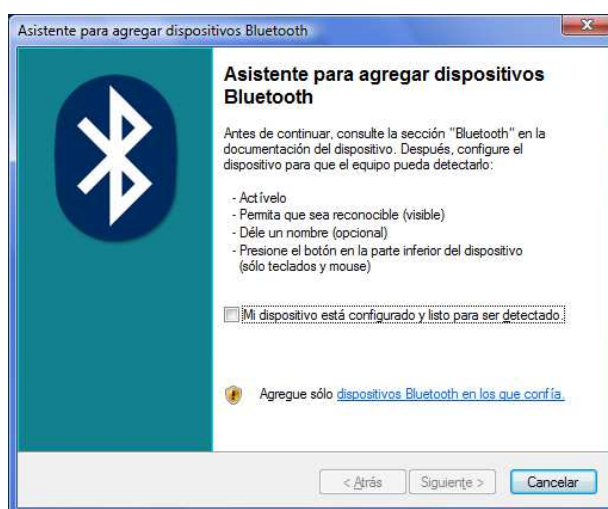


Figura 61: Ventana Agregar Dispositivo Bluetooth

Se marca la opción “Mi dispositivo está configurado y listo para ser detectado” y se pulsa el botón “Siguiente”. A continuación, el sistema realiza una búsqueda de los terminales que encuentra dentro de su radio de alcance Bluetooth y los muestra en una lista como la ilustrada en la siguiente figura.

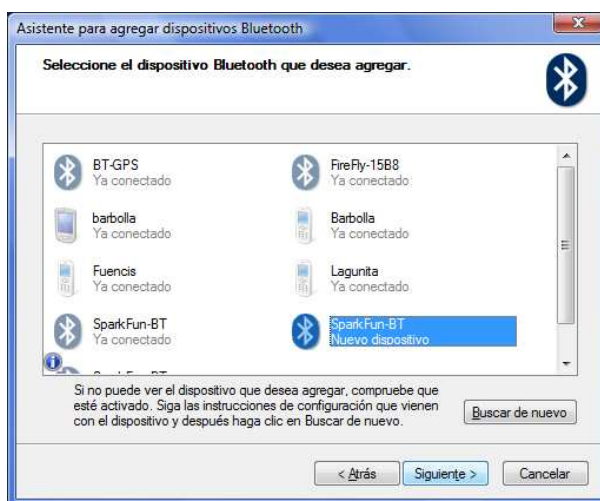


Figura 62: Ventana Seleccionar Dispositivo Bluetooth

Se selecciona el dispositivo que se desea agregar y se pulsa en el botón “Siguiente”. La posterior ventana que se abre es la mostrada debajo.

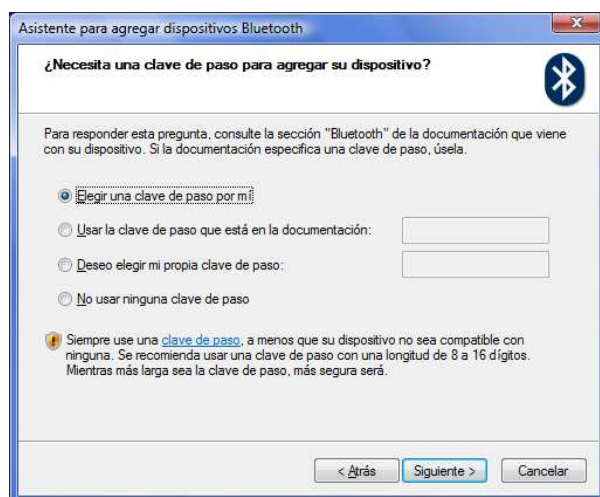


Figura 63: Ventana Clave de Paso Dispositivo Bluetooth

Dependiendo de la configuración de fábrica del dispositivo, comúnmente incluida en la documentación del mismo, se elige una opción acerca de la clave de paso de la conexión. Para los sensores de aceleración se selecciona “Usar la clave de paso

que está en la documentación” e introducir la contraseña “*default*”. Finalmente, se pulsa el botón “Siguiente”.

El sistema establece la conexión mostrando la información de los pasos realizados en una ventana como la siguiente.

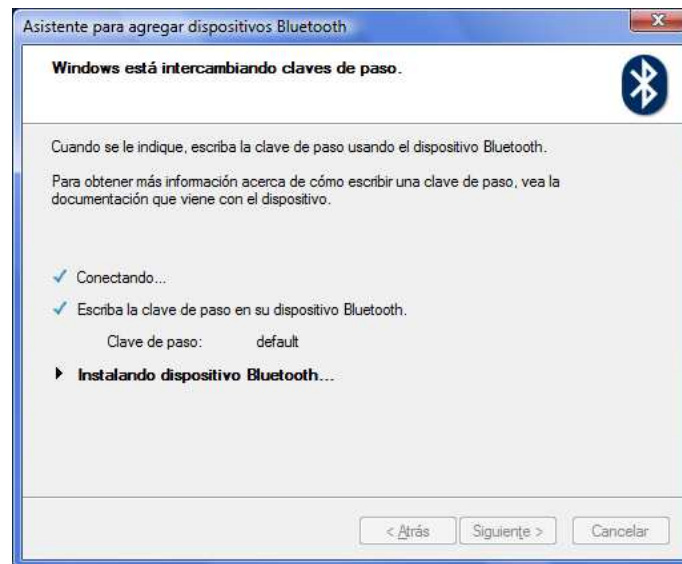


Figura 64: Ventana Pasos Conexión

Finalmente, el sistema nos proporciona dos puertos COM uno de entrada y otro de salida que serán los utilizados en las futuras conexiones. En la siguiente figura se ilustra este proceso.

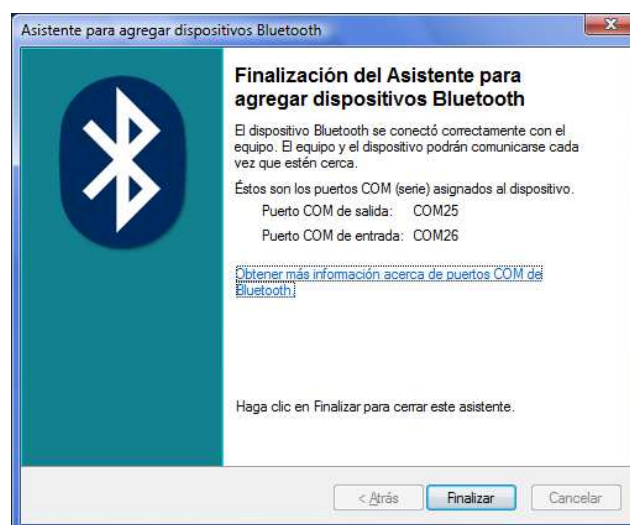


Figura 65: Ventana Puertos COM

El puerto COM de salida indicado es el que va a ser utilizado a continuación. En este ejemplo concreto el asignado es el denominado COM25. A partir de este momento, vamos a hacer uso del terminal. Se abre el programa y se crea una nueva conexión introduciendo un nombre en la ventana siguiente.



Figura 66: Ventana Conexión Terminal

En la siguiente pantalla, ilustrada debajo, se selecciona el puerto anteriormente establecido para el dispositivo, en este caso el COM25, y se pulsa el botón “OK”.



Figura 67: Ventana Puerto Terminal

A continuación es necesario configurar una serie de parámetros. Para ello, se selecciona del menú principal la opción “File”, dentro de ella “Properties” y por último “Configure”. Se deben seleccionar los siguientes valores: en la pestaña “General” la velocidad del puerto 115200 y en la pestaña “Opciones avanzadas” Bits de datos 8, Paridad Ninguna, Bits de parada 1 y modulación Estándar. El estado final de esta configuración se muestra en la imagen de abajo.

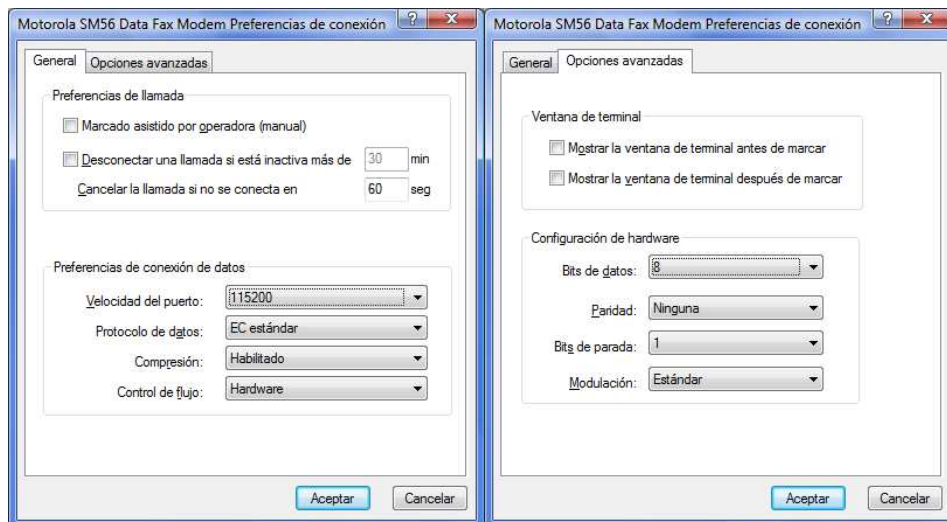


Figura 68: Ventanas Configuración Parámetros Terminal

Si se selecciona la opción “Call” la conexión es establecida y se escriben por la consola los datos recibidos del sensor. En un principio, el menú con las opciones del sensor y a continuación, tras teclear un uno, los datos de posición del sensor. El establecimiento de la conexión también es advertido en el sensor mediante el encendido de una luz verde. Un ejemplo de los datos mostrados en el terminal del ordenador es el siguiente.

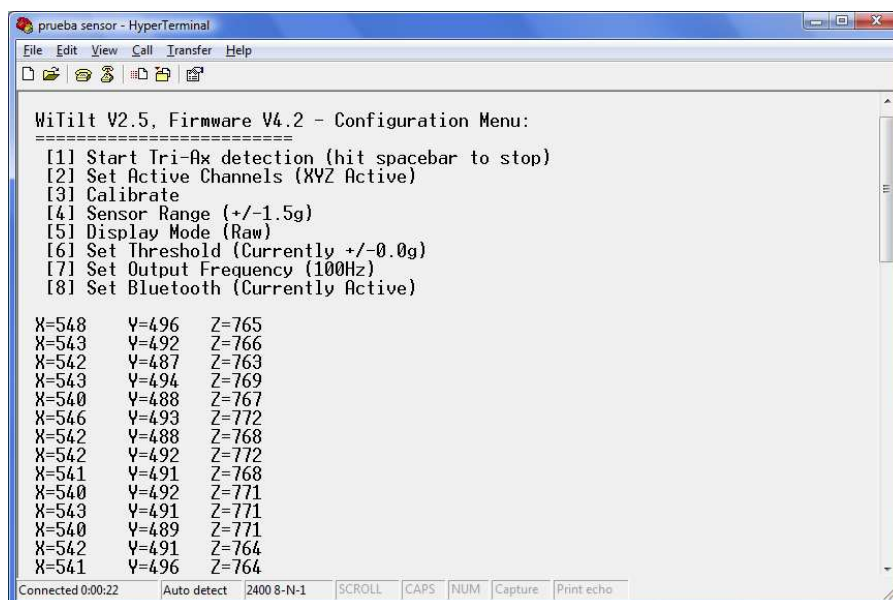


Figura 69: Ventana Datos Terminal

En ocasiones es necesario pulsar el botón Reset de los sensores para que estos respondan tras establecer la conexión y el envío del carácter ‘1’ sea recibido. Para

detener el envío de datos y volver al menú del sensor es necesario pulsar la barra espaciadora. Si se desea finalizar la conexión se selecciona la opción “Disconnect”. Por otro lado, si se desea realizar un cambio en la configuración del sensor, se puede realizar a partir de las opciones mostradas en el menú de la anterior figura. Por ejemplo, para variar la frecuencia de envío de datos del sensor se debe escoger la opción número 7 “Set Output Frequency”.

Si todos los pasos anteriores son realizados de manera correcta y se pueden ver los datos del sensor en el programa del terminal, el estado del sensor es el adecuado para su utilización en el sistema del proyecto.

C Ejemplo Conexión Bluetooth mediante Puertos COM

Este ejemplo ilustra la manera utilizada para establecer conexiones con dispositivos Bluetooth a través de los puertos serie de una computadora. Para la configuración del dispositivo en el ordenador y la obtención de un puerto de conexión es necesario seguir los pasos indicados en el anterior anexo.

El programa de este ejemplo ha sido codificado en lenguaje Java y para su ejecución el único requisito es la instalación de la máquina virtual de dicho lenguaje en la computadora.

C.1 Funcionamiento de la Aplicación

El funcionamiento de la aplicación es el siguiente:

- 1) Se incluyen dos parámetros de configuración en la llamada de la ejecución de la aplicación: el puerto COM por el cual se va a realizar la conexión y el número de bytes o caracteres que se quieren leer.
- 2) El sistema tratará de abrir la conexión por el puerto serie indicado.
- 3) Si la apertura se realiza adecuadamente, se procederá a la lectura del número de bytes indicados como parámetro que se reciban del dispositivo a la vez que van siendo mostrados por pantalla.
- 4) Cuando la lectura finalice el sistema cerrará la conexión y la ejecución concluirá.

C.2 Ejecución de la Aplicación

Para la ejecución concreta de la aplicación que se va a exponer a continuación, se ha empleado, conjuntamente con el ordenador portátil usado a lo largo del desarrollo del proyecto, un dispositivo antena GPS con conexión Bluetooth. En concreto se trata del modelo cuya imagen se muestra en la siguiente figura.



Figura 70: Dispositivo GPS

El paso inicial para su ejecución consiste en la compilación del fichero fuente mediante el siguiente comando “javac EjemploSerial.java”. Si el sistema tiene instalada y configurada correctamente la Máquina Virtual de Java, automáticamente se genera el fichero ejecutable. Mediante el comando “java EjemploSerial” el programa inicia su funcionamiento. Son necesarios dos parámetros de configuración en la llamada al sistema: el identificador del puerto por el que se quiere establecer la conexión y el número de caracteres que se desea leer.

El correcto funcionamiento de la aplicación se ilustra en la siguiente imagen:

```

C:\Windows\system32\cmd.exe

C:\Users\David\Desktop\ejemplosJava>javac EjemploSerial.java
C:\Users\David\Desktop\ejemplosJava>java EjemploSerial COM23 1000

Conectamos el dispositivo por el puerto COM23
Dispositivo conectado
Leyendo los primeros 1000 bytes
$GPGGA,000000.000,0000.0000,S,00000.0000,W,0.00,00.0,0.0,M,0.0,M,,*52
$GPGSA,A,1,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,000000.000,U,0000.0000,S,00000.0000,W,0.00,0.00,220899,.,A*7E
$GPGGA,000000.000,0000.0000,S,00000.0000,W,0.00,00.0,0.0,M,0.0,M,,*52
$GPGSA,A,1,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,000000.000,U,0000.0000,S,00000.0000,W,0.00,0.00,220899,.,A*7E
$GPGGA,000000.000,0000.0000,S,00000.0000,W,0.00,00.0,0.0,M,0.0,M,,*52
$GPGSA,A,1,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,000000.000,U,0000.0000,S,00000.0000,W,0.00,0.00,220899,.,A*7E
$GPGGA,000000.000,0000.0000,S,00000.0000,W,0.00,00.0,0.0,M,0.0,M,,*52
$GPGSA,A,1,,,,,,,,,0.0,0.0,0.0*30
$GPRMC,000000.000,U,0000.0000,S,00000.0000,W,0.00,0.00,220899,.,A*7E
$GPGGA,000000.000,0000.0000,S,00000.0000,W,0.00,00.0,0.0,M,0.0,M,,*52
$GPGSA,A,1,,,,,,,,,0.0,0.0,0.0*30
Lectura finalizada
C:\Users\David\Desktop\ejemplosJava>
  
```

Figura 71: Ejecución ejemplo conexión Bluetooth mediante puertos COM

Como se aprecia en la anterior figura, el programa ha funcionado correctamente. La conexión se ha establecido por el puerto “COM23” y se han mostrado por pantalla los primeros 1000 caracteres recibidos del dispositivo.

El fichero con el código fuente del programa, EjemploSerial.java, se encuentra debidamente comentado y disponible para su uso en el CD adjunto en la entrega del proyecto en la carpeta “ejemplos” y la subcarpeta “Puertos_COM”.

D Ejemplo Práctico de Uso del API de Java para Bluetooth

En este capítulo se muestra un ejemplo de programación de aplicaciones que utilizan conexiones Bluetooth codificado en lenguaje Java y mediante el API para Bluetooth, el denominado JSR-82.

El objetivo del programa es sencillo y muy general. Sin embargo, a partir de su código fuente se pueden generar multitud de aplicaciones que trabajen con conexión Bluetooth. En este caso, la conexión simplemente se establece, pero a partir de ahí podría comenzar cualquier intercambio de datos entre ambos terminales.

D.1 Funcionamiento de la Aplicación

El funcionamiento de la aplicación es el siguiente:

- 1) El programa realiza una comprobación de que el Bluetooth del dispositivo local está disponible y preparado para utilizar.
- 2) A continuación, inicia una búsqueda de los dispositivos que se encuentran en su radio de alcance y muestra los encontrados en una lista.
- 3) El usuario escoge uno y el sistema comienza una búsqueda de los servicios Bluetooth que se encuentran disponibles para el dispositivo elegido.
- 4) A su finalización, los servicios son enumerados de igual manera y se permite al usuario escoger uno de ellos para establecer una conexión.
- 5) El programa establece la conexión entre ambos dispositivos.
- 6) Finalmente, se permite al usuario salir de la aplicación.

D.2 Ejecución de la Aplicación

Esta aplicación ha sido instalada para su evaluación en un teléfono móvil de la marca Nokia y modelo 6210. Además, se ha utilizado una serie de dispositivos Bluetooth que se han encendido y situado en el rango de alcance del teléfono para que el programa los encuentre los cuales serán enumerados posteriormente.

A continuación se muestran unas imágenes que ilustran una correcta ejecución del mismo.



Figura 72: Ejemplo API Java para Bluetooth

En la primera pantalla se observa que la comprobación del estado del Bluetooth del dispositivo se ha realizado correctamente. En la segunda se puede ver la lista con los identificadores de los dispositivos encontrados por el teléfono móvil, en este orden el ordenador portátil, la PDA, un mando de la videoconsola Wii y la antena GPS. En la tercera se observa el servicio ofrecido por el dispositivo seleccionado, en este caso la antena GPS. En la última se incluye la información sobre el dispositivo GPS, su clase y su tipo, y se ofrece la posibilidad de conectar y de salir de la aplicación.

El fichero ejecutable de la aplicación, Bluemoved.jar, así como el código fuente de la aplicación, se encuentra almacenado en el CD adjunto en la entrega del proyecto en la carpeta “ejemplos” y la subcarpeta “J2ME”.

E Ejemplo de Conexión mediante el Puerto Serie de Palm

Para comprobar su funcionamiento, se ha realizado un ejemplo práctico de conexión Bluetooth mediante el puerto serie del dispositivo Palm Tungsten E2. Consiste en un programa en el que se establece una conexión con un sensor de aceleración como los que se van a emplear en este proyecto. Para ello, se accede a las librerías, proporcionadas en el sitio web oficial de programación de este tipo de dispositivos, que permiten manejar el puerto serie [30].

La aplicación está disponible para su uso en el CD adjunto. De igual modo, se encuentra el código fuente del programa. Ambos se hallan alojados en el directorio “ejemplos” y el subdirectorio “PalmWiTilt”.

E.1 Funcionamiento de la Aplicación

El funcionamiento de la aplicación es el siguiente:

- 1) El programa comprueba si la conexión Bluetooth del dispositivo local está activa y lista para ser utilizada.
- 2) Se inicia una búsqueda de dispositivos Bluetooth al alcance y se muestran en una lista para que el usuario elija el que desee.
- 3) Se conecta con el dispositivo escogido a través del puerto serie.
- 4) Por la conexión creada, se manda al dispositivo el comando de comienzo de envío de datos.
- 5) Conforme se van recibiendo los datos a través del puerto serie, estos son leídos y mostrados por pantalla.
- 6) Cuando el usuario lo desea se finalizan la lectura de datos y la conexión por el puerto serie. A continuación, automáticamente finaliza el programa.

E.2 Ejecución de la Aplicación

En la ejecución se han empleado el dispositivo móvil Palm Tungsten E2 y un sensor de aceleración WiTiltv2.5. Ambos son los mismos modelos que han sido empleados para la solución final del proyecto.

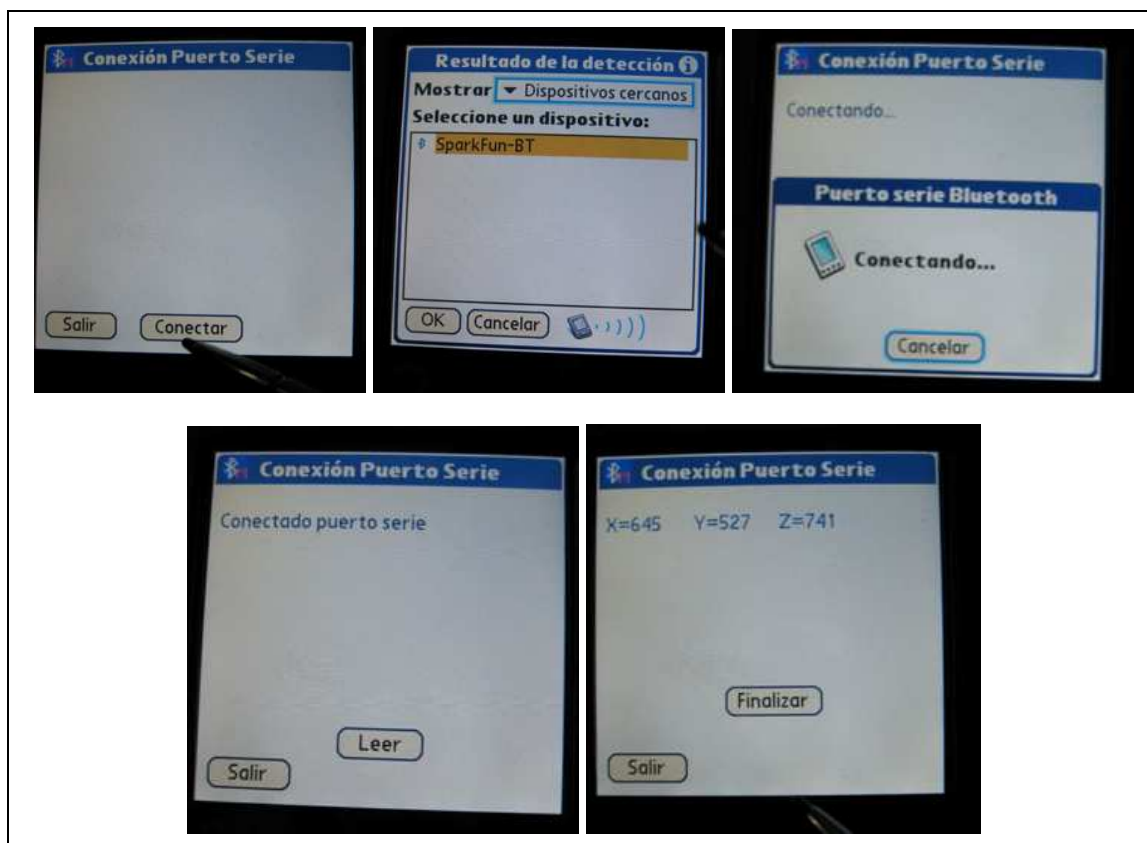


Figura 73: Ejemplo conexión puerto serie Palm

F Ejemplo Conexión mediante Sockets Bluetooth de Palm

Este ejemplo consiste en el establecimiento de conexión mediante sockets Bluetooth de la Palm Tungsten E2 con otro dispositivo. En concreto, se ha utilizado un mando de la conocida videoconsola Wii de Nintendo cuya imagen se observa en la siguiente figura.



Figura 74: Mando videoconsola Wii

El objetivo de este ejemplo es comprobar el funcionamiento de dicha conexión tratando de obtener los valores de los botones que se pulsan en el mando. Este dispositivo utiliza tecnología Bluetooth para conectar con la videoconsola mediante un protocolo HID. Funciona como un emisor de datos que la computadora recibe y procesa. Nuestra intención es la de simular el un reducido comportamiento de la misma. Concretamente, la comunicación con el mando y la muestra de los datos recibidos.

Tanto el fichero ejecutable de la aplicación como su código fuente, están accesibles en el CD adjunto. Todos ellos se almacenan en la carpeta “ejemplos” y el subdirectorio “PalmWii”.

F.1 Funcionamiento de la Aplicación

El funcionamiento de la aplicación está resumido en los siguientes pasos:

- 1) El programa comprueba si la conexión Bluetooth del dispositivo local está activa y lista para ser utilizada.
- 2) Si es así, se inicia la búsqueda del dispositivo y se muestra una lista con los dispositivos Bluetooth al alcance para que el usuario elija.
- 3) Se conecta con el dispositivo escogido.

- 4) Para la comunicación, se crea un socket L2CAP de salida de datos el cual se conecta con el otro dispositivo.
- 5) De igual manera, se crea un socket L2CAP de entrada de datos el cual se conecta con el otro dispositivo.
- 6) Finalmente, se van mostrando por la pantalla del dispositivo los botones que el usuario va pulsando en el mando.

F.2 Ejecución de la Aplicación

Como se ha comentado anteriormente, para realizar el ejemplo se ha empleado un mando de la videoconsola Wii [31]. Mediante la aplicación EjemploBluetooth.java expuesto previamente en el Anexo D, se ha obtenido la información necesaria acerca de este dispositivo para este ejemplo, la cual se encuentra recogida en la siguiente tabla.

Información	Valor
Nombre del dispositivo	Nintendo RVL-CNT-01
Dirección del dispositivo	00:1F:32:9E:E0:EF
Major Device Class	1280 (Periférico)
Minor Device Class	4 (Mando de juego)

Tabla 47: Información acerca del mando de la videoconsola Wii

Cabe añadir que por su especificación se sabe que el dispositivo acepta conexiones mediante el protocolo Bluetooth L2CAP.

La aplicación ha sido ejecutada en el dispositivo Palm Tungsten E2 empleado en la resolución de este proyecto. Las siguientes imágenes ilustran el correcto funcionamiento de esta aplicación.

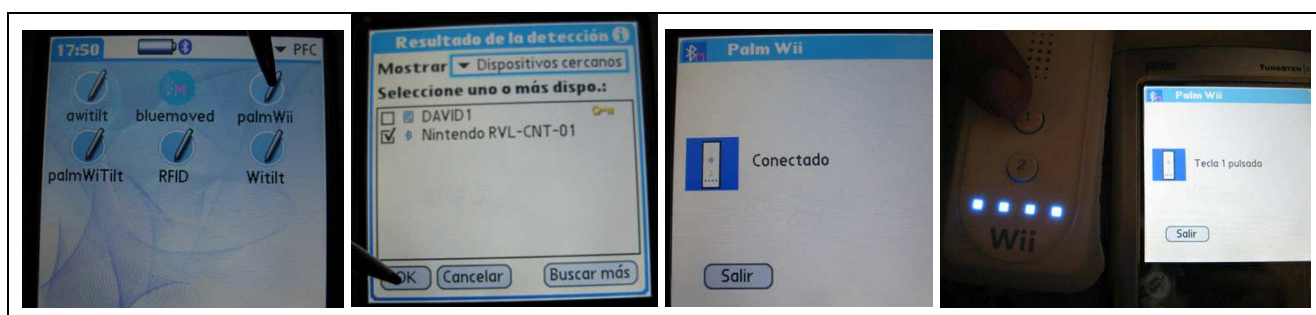


Figura 75: Ejemplo conexión mediante sockets Bluetooth

En la primera imagen se observa el icono de la aplicación denominada “palmWii”. A continuación se muestra cómo el dispositivo ha encontrado el mando, cuyo nombre Bluetooth es “Nintendo RVL-CNT-01”. En la siguiente pantalla el programa muestra que se ha establecido correctamente la conexión. En la última imagen se observa cómo aparece automáticamente reflejado en la pantalla de la PDA el botón 1 que ha sido pulsado en el mando.

G Contenido del CD

La entrega del proyecto fin de carrera se ha realizado en formato electrónico, en concreto, se ha grabado en un disco compacto el conjunto de archivos que conforman el contenido completo de este trabajo.

Este CD se ha organizado en los siguientes directorios:

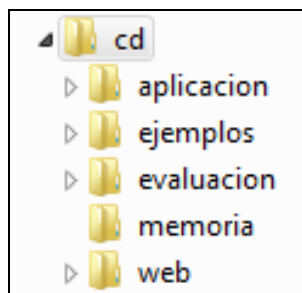


Figura 76: Organización del CD

El directorio “aplicacion” contiene los ficheros de la fase de programación. Por un lado se encuentra el fichero ejecutable de la aplicación “Bluemoved2.prc”. Por otro, se encuentran los archivos con el código fuente completo y comentado.

De igual manera, en el directorio “ejemplos” se hallan almacenados los ficheros fuente y ejecutable de cada uno de los ejemplos comentados en los anexos anteriores de este documento. Están estructurados por carpetas cuyos nombres se corresponden con los ejemplos y son: J2ME, Palm_Wii, Palm_WiTilt, Puertos_COM.

La carpeta “evaluación” alberga todos los ficheros de texto de la fase de evaluación referenciados en este documento y obtenidos a partir de las diferentes pruebas realizadas para la comprobación del programa. Al igual que en este documento se hallan ordenados por la funcionalidad de los mismos en las carpetas: calidad, funcionamiento, hyperterminal y rendimiento.

En el directorio memoria se encuentra almacenado el documento presente denominado “memoriaPFC.doc”.

Por último, la carpeta web contiene los ficheros necesarios para ofrecer una presentación del proyecto en formato de página web cuyo contenido se carga a partir del archivo index.html.

Referencias

[1] Aplicaciones de los dispositivos móviles

Orientación de los dispositivos móviles a aplicaciones para usuarios y empresas

<http://www.estoemarketing.com/Comunicacion%20alternativa/MCommerce.pdf>

[2] Palm Tungsten E2

Descripción del modelo en la página oficial de la marca.

<http://www.palm.com/es/es/products/handhelds/e2/>

[3] Historia Palm Tungsten E2

Evolución de los productos de la marca hasta el modelo Palm Tungsten E2

http://en.wikipedia.org/wiki/Tungsten_e2

[4] Características Tungsten E2

Características hardware y software del dispositivo Palm Tungsten E2

http://www.pdaexpertos.com/modelos_pda/palmone_tungsten_e2.shtml

[5] Definición de acelerómetro

Descripción de los fundamentos y características de los sensores de aceleración

<http://es.wikipedia.org/wiki/Acelerometro>

[6] Acelerómetros para robots

Definición de acelerómetros y sus aplicaciones en el campo de la robótica

<http://turan.uc3m.es/uc3m/dpto/IN/dpin04/ISL/sensores.pdf>

[7] Sensor WiTilt v2.5

Información completa sobre el sensor de aceleración WiTilt v2.5

http://www.robotshop.us/PDF/WiTilt_V2_5.pdf

[8] Sensor WiTilt v3.0

Información completa sobre el sensor de aceleración WiTilt v3.0

http://www.sparkfun.com/commerce/product_info.php?products_id=8563

[9] Aplicación acelerómetros

Aplicación en la que se emplea un sensor de aceleración para el control de movimientos
<http://bid.berkeley.edu/files/papers/47170019.PDF>

[10] Aplicación gráfica WiTilt v2.5

Aplicación gráfica en la que se emplea un sensor de aceleración WiTilt v2.5
<http://www.youtube.com/watch?v=2KKXuZkuZs8&hl=es>

[11] Aplicación WiTilt v2.5

Aplicación de salto al vacío en la que se emplea un sensor de aceleración WiTilt v2.5
<http://www.sparkfun.com/commerce/news.php?id=194>

[12] Definición sistema RFID

Definición de los fundamentos y las características de los sistemas RFID
<http://es.wikipedia.org/wiki/RFID>

[13] Sensor RFID

Descripción del dispositivo RFID utilizado en el proyecto
<http://www.sonmicro.com/125/d125.php>

[14] Ampliaciones RFID

Ampliaciones realizadas al sensor RFID del proyecto
<http://www.cc.gatech.edu/ccg/resources/wearableRFID.html#ss20>

[15] Conector Bluetooth RFID

Descripción del dispositivo Bluetooth usado para conectar el sensor RFID del proyecto
<http://www.sonmicro.com/125/d125.php>

[16] Aplicación RFID

Aplicación de reconocimiento de colores mediante el uso de un sensor RFID
<http://www.cc.gatech.edu/~eugenem/pubs/medynskiy-gov-mazalek-minnen%20wearable%20rfid%20for%20play.pdf>

[17] Estudio de las capacidades Bluetooth

Proyecto Fin de Carrera sobre las capacidades Bluetooth.
Autor: García Morón, Begoña
Biblioteca Politécnica Universidad Carlos III de Leganés PFC 02003, 2004

[18] Análisis de las prestaciones de la tecnología Bluetooth

Proyecto Fin de Carrera sobre las prestaciones Bluetooth.
Autor: Téllez García Moreno, Virginia
Biblioteca Politécnica Universidad Carlos III de Leganés PFC 02460, 2004

[19] Bluetooth application developer's guide:the short range interconnect solution

Autor: Kammer, David

Editor: Syngress

ISBN: 9781928994428

[20] Bluetooth SIG

Página oficial del grupo de investigación especial de Bluetooth

<http://spanish.bluetooth.com/Bluetooth/SIG/>

[21] Versiones Bluetooth

Información acerca de las versiones existentes de la especificación Bluetooth

http://es.wikipedia.org/wiki/Perfil_Bluetooth

[22] Comparación Bluetooth

Comparación de Bluetooth con las tecnologías inalámbricas existentes

<http://spanish.bluetooth.com/Bluetooth/Technology/Works/Compare/>

[23] Protocolos Bluetooth

Descripción de la pila de protocolos de la tecnología inalámbrica Bluetooth

<http://www.electronicafacil.net/tutoriales/Protocolos-Bluetooth.php>

[24] Perfiles Bluetooth

Descripción de los perfiles de la tecnología inalámbrica Bluetooth

http://spanish.bluetooth.com/Bluetooth/Technology/Works/Profiles_Overview.htm

[25] API Java para Bluetooth

Especificación estándar del JSR82, el API de Java para Bluetooth

<http://jcp.org/en/jsr/detail?id=82>

[26] PalmOS Programming Bible

Autor: Foster, Lonnon R.

Editor: Hungry Minds

ISBN: 0764546767

[27] Palm programming

Autor: Rhodes, Neil

Editor: O'Reilly

ISBN: 1565925254

[28] Tutorial Palm

Página web destinada para la iniciación en la programación en dispositivos Palm

<http://mobile.eric-poncet.com/palm/tutorial.html>

[29] Ayuda Bluetooth PalmOS

Documento con la ayuda necesaria para el manejo del Bluetooth de un dispositivo Palm
http://bluetoothpages.com/doc/palm_bluetooth.pdf

[30] Librerías Palm

Documentación oficial para la programación de los dispositivos Palm
<http://www.accessdevnet.com/index.php/GarnetDocs/Garnet-OS.html>

[31] Mando Wii

Información técnica sobre el mando de la videoconsola Wii
<http://wiibrew.org/w/index.php?title=Wiiimote>

[32] GarnetOS Development Suite

Descarga oficial de la herramienta de desarrollo para dispositivos Palm
<http://www.accessdevnet.com/index.php/ACCESS-Tools/View-category.html>

[33] Fundación Eclipse

Información sobre la fundación Eclipse
<http://www.eclipse.org/org/>

[34] Fundación Apache

Información sobre la fundación Apache
<http://www.apache.org/>

[35] Hyperterminal

Página de descarga de la herramienta Hyperterminal
<http://www.softonic.com/s/hyperterminal>

Acrónimos

ACL	<i>Asynchronous Connectionless Link</i> : enlace físico de banda base no orientado a conexión asíncrono
API	<i>Application Programming Interface</i> : interfaz de programación de una aplicación
BBDD	Bases de Datos
BIN	<i>BIN</i> ary: binario
CD	<i>Compact Disc</i> : disco compacto
GIAC	<i>General Inquiry Access Code</i> : Código de acceso de descubrimiento general
GPS	<i>Global Positioning System</i> : sistema de posicionamiento global
GPRS	<i>General Packet Radio Service</i> : servicio general de paquetes vía radio extensión de GSM.
GSM	<i>Global System for Mobile Communications</i> : sistema estándar para la comunicación mediante teléfonos móviles que incorporan tecnología digital.
HID	<i>Human Interface Device</i> : interfaz de control de equipo.
IAC	<i>Inquiry Access Code</i> : código de acceso de búsqueda.
IDE	<i>Integrated Development Enviroment</i> : entorno de desarrollo integrado.
IEEE	<i>Institute of Electrical and Electronics Engeneers</i> : instituto de ingenieros eléctricos y electrónicos.
IP	<i>Internet Protocol</i> : protocolo no orientado a conexión usado para la comunicación de datos a través de una red de paquetes conmutados.
IR	<i>InfraRed</i> : infrarrojos

IrDA	<i>Infrared Data Association</i> : asociación de datos por infrarrojos
ISM	<i>Industrial Scientific and Medical</i> : banda de frecuencia dedicada a la investigación.
LAN	<i>Local Area Network</i> : red de área local.
LMP	<i>Link Manager Protocol</i> : protocolo de gestión de enlaces
L2CAP	<i>Logical Link Control and Adaptation Protocol</i> : protocolo de adaptación y control de enlace lógico
MAC	<i>Media Access Control</i> : control de acceso al medio.
MPEG	<i>Moving Picture Expert Group</i> : grupo de expertos en imágenes móviles
MP3	<i>MPEG Audio Layer 3</i> : formato de audio digital comprimido
MVC	<i>Model View Controller</i> : patrón de arquitectura software modelo-vista-controlador.
OBEX	<i>OBject EXchange</i> : protocolo de intercambio de objetos.
OS	<i>Operating System</i> : sistema operativo
OSI	<i>Open System Interconnection</i> : sistema que define la torre de protocolos más usada en sistemas fijos.
PDA	<i>Personal Digital Assistant</i> : asistente digital personal
PC	<i>Personal Computer</i> : ordenadores personales
PPP	<i>Point to Point Protocol</i> : protocolo de Internet que proporciona un modo de transportar datagramas de distintos protocolos sobre enlaces punto a punto.
RAM	<i>Random Access Memory</i> : memoria de acceso aleatorio
RFCOMM	<i>Radio Frequency Communication</i> : protocolo de emulación de puerto serie
RFID	<i>Radio Frequency IDentification</i> : identificación por radiofrecuencia.
ROM	<i>Read Only Memory</i> : memoria de solo lectura.
SCO	<i>Synchronous Connected Oriented</i> : enlace físico de banda base síncrono orientado a conexión.
SD	<i>Secure Digital</i> : formato de tarjeta de memoria flash
SDIO	<i>Secure Digital Input Output</i> : tarjeta SD de almacenamiento de entrada/salida

SDK	<i>Software Development Kit</i> : conjunto de herramientas de desarrollo que permite a un programador crear aplicaciones para un sistema concreto.
SDP	<i>Service Discover Protocol</i> : protocolo de descubrimiento de servicios
SIG	<i>Special Interest Group</i> : grupo de interés especial.
SMS	<i>Short Message Service</i> : servicio de mensajes cortos
SPP	<i>Service Port Profile</i> : perfil del puerto serie empleado para definir su emulación a través de RFCOMM.
TCP	<i>Transport Control Protocol</i> : protocolo de Internet que ofrece un transporte fiable.
TCP/IP	Pila de protocolos de nivel de red y transporte de la torre OSI.
TCS	<i>Telephony Control Specification</i> : Especificación del protocolo de control de telefonía de Bluetooth.
TFT	<i>Thin Film Transistor</i> : transistor de película fina
UDP	<i>User Datagram Protocol</i> : protocolo de transporte de datagramas no fiable.
UML	<i>Unified Modeling Language</i> : lenguaje unificado de modelado.
USB	<i>Universal Serial Bus</i> : bus universal en serie
UUID	<i>Universal Unique Identifier</i> : identificador único universal.
UWB	<i>Ultra WideBand</i> : tecnología de radio que usa un ancho de banda superior a 500 MHz.
WAE	<i>Wireless Application Environment</i> : entorno de aplicación inalámbrica.
WAP	<i>Wireless Application Protocol</i> : protocolo de aplicaciones inalámbricas.
Wi-Fi	<i>Wireless Fidelity</i> : denominación de una red inalámbrica de alta velocidad.
WPAN	<i>Wireless Personal Area Networks</i> : redes inalámbricas de área personal